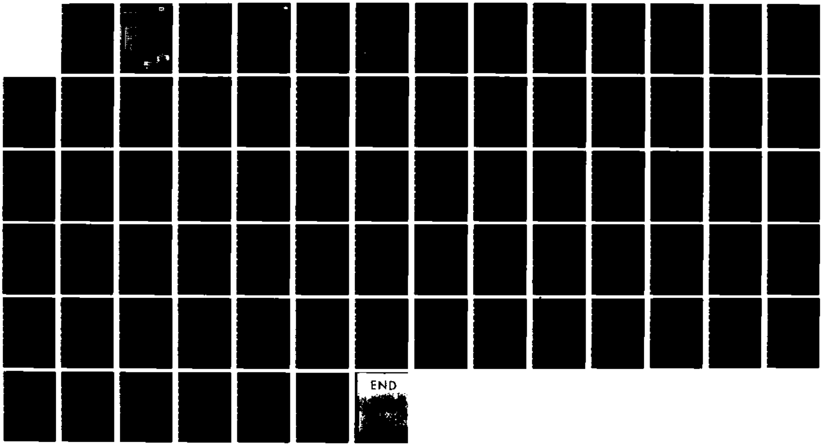
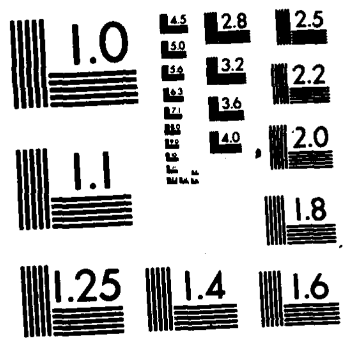
AD-A134 893 COMPUTER MODELS FOR TWO-DIMENSIONAL TRANSIENT HEAT CONDUCTION(U) COLD REGIONS RESEARCH AND ENGINEERING LAB HANOVER NH M R ALBERT APR 83 CRREL-83-12 1/1

UNCLASSIFIED F/G 9/2 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

CRREL
REPORT 83-12

A134893

US Army Corps of Engineers
Cold Regions Research & Engineering Laboratory

Computer models for two-dimensional transient heat conduction

For conversion of SI metric units to U.S./British customary units of measurement consult ASTM Standard E380, Metric Practice Guide, *published by the American Society for Testing and Materals, 1916 Race St., Philadelphia, Pa. 19103.*

Cover: Finite difference grid of nodes for the semi-infinite corner problem.

CRREL Report 83-12

April 1983

Computer models for two-dimensional transient heat conduction

Mary Remley Albert

Prepared for
OFFICE OF THE CHIEF OF ENGINEERS

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CRREL Report 83-12	2. GOVT ACCESSION NO. A134893	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) COMPUTER MODELS FOR TWO-DIMENSIONAL TRANSIENT HEAT CONDUCTION		5. TYPE OF REPORT & PERIOD COVERED 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Mary Remley Albert		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS U.S. Army Cold Regions Research and Engineering Laboratory Hanover, New Hampshire 03755		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS DA Project 4A762730AT42, Technical Area D, Work Unit 017
11. CONTROLLING OFFICE NAME AND ADDRESS Office of the Chief of Engineers Washington, D.C. 20314		12. REPORT DATE April 1983 13. NUMBER OF PAGES 74
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Centralized heating systems
Computerized simulation
Computer models
Heat pipes
Heat transfer

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper documents the development and verification of two finite difference models that solve the general two-dimensional form of the heat conduction equation, using the alternative-direction implicit method. Both can handle convective, constant flux, specified temperature and semi-infinite boundaries. The conducting medium may be composed of many materials. The first program, ADI, solves for the case where no change of state occurs. ADIPC solves for the case where a freeze/thaw change of phase may occur, using the apparent heat capacity method. Both models are verified by comparison to analytical results.

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

PREFACE

This report was prepared by Mary Remley Albert, Mathematician, of the Applied Research Branch, Experimental Engineering Division, U.S. Army Cold Regions Research and Engineering Laboratory. The work was sponsored by DA Project 4A762730AT42, *Design, Construction and Operations Technology for Cold Regions,* Technical Area D, *Cold Regions Design and Construction,* Work Unit 017, *Heat Distribution Systems in Cold Regions.*

The author thanks Gary Phetteplace of CRREL for his support and technical review. Also, the author thanks Dr. Devinder Sodhi and Dr. Kevin O'Neill, both of CRREL, for their technical review.

CONTENTS

	Page
Abstract	i
Preface	ii
Introduction	1
Finite differences applied to heat transfer	2
Heat conduction equation	2
Boundary conditions	6
Phase change	14
Computer program	16
ADDATA, the data subroutine	16
TRIDIG, the matrix solver	16
ISOTHM, the isotherm finder	17
ADI, main program	17
ADIPC, main program	18
Verification of ADI	18
Comparison of ADI with analytical results	18
Comparison of ADI with experimental results	23
Verification of ADIPC	25
Comparison of ADIPC with analytical results—the Neumann solution	25
Comparison of ADIPC with analytical results—two-dimensional phase change verification	27
User instruction for ADI	29
User instruction for ADIPC	29
Conclusions	30
Literature cited	30
Appendix A: Program INFSUM and sample input and output for program ADI	31
Appendix B. Program ADIPC and sample input and output	37

ILLUSTRATIONS

Figure		
1.	A simple finite difference grid	2
2.	A node on the interior of the grid	3
3.	A node on a right-hand boundary	6
4.	A node on a semi-infinite boundary	8
5.	An interior node adjacent to the semi-infinite boundary	9
6.	A node on a corner of the grid	10
7.	A semi-infinite corner	11
8.	A semi-infinite side node adjacent to a semi-infinite corner	12
9.	A node on a semi-infinite corner	13
10.	Tridiagonal matrix equation	17
11.	Semi-infinite corner problem	18
12.	Finite difference grid for the semi-infinite corner problem	19
13.	Locations of 35°, 30° and 25°C isotherms from ADI and from the analytical solution for the semi-infinite corner problem	19
14.	Finite rectangle problem	20

Figure		Page
15.	Locations of 24°, 28°, 32°, 36° and 40°C isotherms from ADI and analytical solutions	21
16.	One-dimensional semi-infinite problem	22
17.	Thermocouple locations in the experimental setup	23
18.	Surface temperatures from experiment used in ADI runs	24
19.	Results of two ADI runs using different conductivities plotted against experimental results	25
20.	Comparison of ADIPC with Neumann solution	27
21.	Comparison of ADIPC with radial analytic freezing solution	28

TABLES

Table		
1.	Comparison of ADI results to analytical results for the one-dimensional semi-infinite problem	22

COMPUTER MODELS FOR TWO-DIMENSIONAL TRANSIENT HEAT CONDUCTION

Mary Remley Albert

INTRODUCTION

Most major Army installations are heated with central heat distribution systems. The thermal regimes around buried distribution systems are of interest for several reasons, not the least of which is the estimated millions of dollars lost annually from these systems (Phetteplace et al. 1981). Also, when replacing damaged insulation or installing insulation in a new system, it is desirable to know what an optimum balance is between the initial cost of insulation and the continued operating cost of heat losses. In addition, freezing and thawing of the ground around a buried distribution system could damage it, through loss of support and settlement, for example.

Analysis of the thermal regime involves the solution of the heat conduction equation for situations with complicated geometries and a variety of boundary conditions. Usually, these partial differential equations cannot be solved analytically; we must resort to numerical methods. One long-established method is that of finite differences, a relatively straightforward numerical method used successfully to solve a variety of differential equations. A finite difference computer program, set up in a general form to solve the heat conduction equation under a variety of geometries and boundary conditions, provides a powerful tool with which the engineer can assess problems in conductive heat transfer.

The objective of this paper is to document the development and verification of two general two-dimensional finite difference computer programs that were written to model time-varying heat conduction in a medium composed of many materials. The first program, ADI, solves for the general case in which no change of state occurs in the conducting medium. The second program, ADIPC, is an adaptation of ADI that includes the effects of phase change in the conducting medium. The programs are able to handle convective, constant flux, specified temperature and semi-infinite boundaries. Material properties such as thermal conductivity, density and specific heat are allowed to vary with time or temperature. The programs are relatively easy to use, that is, they are easily set up for new conduction problems by those who have a background knowledge of FORTRAN.

The computer programs were written and programmed in FORTRAN by the author on CRREL's PRIME 400 computer.

FINITE DIFFERENCES APPLIED TO HEAT TRANSFER

Finite differences are commonly used in the numerical solution of partial differential equations. The procedure involves the replacement of differentials by differences, and is best illustrated by construction of the so-called finite difference grid. Let us examine a grid that is set up in two dimensions to model heat conduction (Fig. 1). The two dimensions in this case represent spatial independent variables, x and y. Each point of the grid is called a node and represents the area enclosed by the square around it (with unit depth). $T(x, y)$ represents the temperature of node x, y. For each node, the temperature and material properties are assumed uniform for the region it represents. By specifying the initial temperatures at each of the nodes, the nodal thermal properties and the boundary conditions, we can solve the heat conduction equation to determine the temperatures of the nodes at later times.

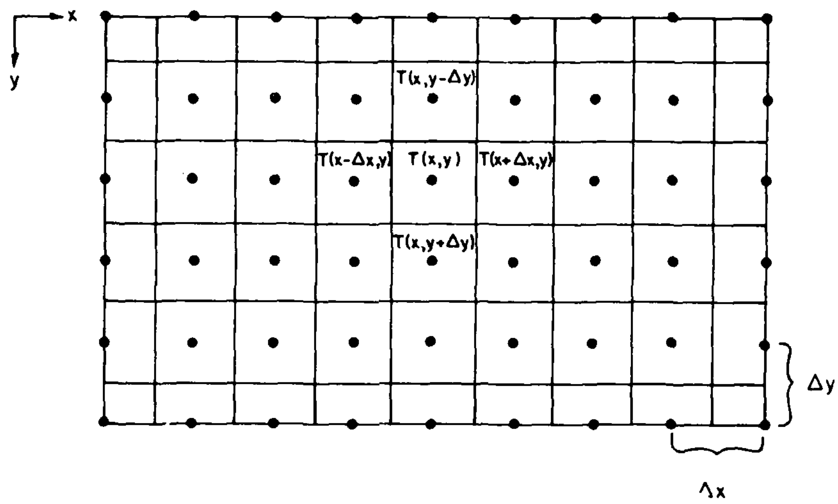


Figure 1. A simple finite difference grid.

Heat conduction equation

The equation governing transient heat conduction in two dimensions is

$$\frac{\partial}{\partial x}\left(k\,\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\,\frac{\partial T}{\partial y}\right)=\rho C_p\,\frac{\partial T}{\partial t} \tag{1}$$

where x and y = spatial variables

T = temperature

t = time

k = thermal conductivity (this may vary over the spatial and time domains)

ρ = density

C_p = specific heat.

The nonlinearities in this partial differential equation that arise from the inclusion of the phase change condition will be discussed in the *Phase Change* section.

Partial differential equations may be expressed in finite difference form either by a Taylor's series expansion about a point or by physical considerations, such as a heat flow balance in the case of the heat conduction problem. For problems involving variable thermal properties or complicated boundary conditions, the heat balance approach is the simplest.

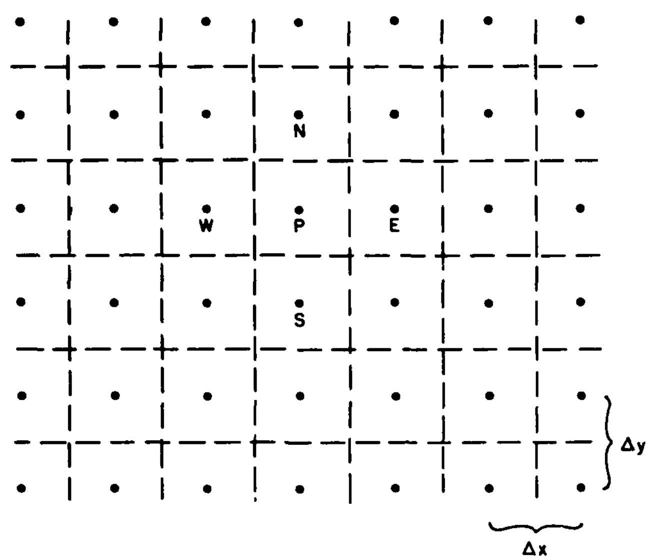


Figure 2. A node on the interior of the grid.

Consider the control volume represented by a node on the interior of the grid, illustrated in Figure 2. The heat flow Q into the control volume from any adjacent node may be calculated as follows:

$$Q = \frac{k}{L} A\Delta T \tag{2}$$

where k = thermal conductivity
L = distance over which the heat flow occurs
A = area (per unit depth) perpendicular to the direction of heat flow
ΔT = change in temperature between the two nodes.

Examine the heat flow from node N to node P. The conductance k/L is the reciprocal of the resistance between the two nodes. The resistance between the two nodes R_{NP} is simply the sum of the individual resistances,

$$R_{NP} = R_N + R_p\,.$$

Each resistance is the distance divided by the material conductivity,

$$R_{NP} = \frac{\Delta y/2}{k_N} + \frac{\Delta y/2}{k_p}$$

then

$$\frac{k}{L} = \frac{1}{R_{NP}} = \frac{2\,k_N k_P}{\Delta y(k_N + k_p)}\,. \tag{3}$$

In the computer program it is assumed that, for nodes not on a boundary of the grid, Δy and Δx are uniform and equal throughout the grid. Let the nodal spacing $\Delta x = \Delta y = \Delta s$. Following the form of eq 2, we find that the heat flows into the region associated with node P from nodes N, W, S and E may be calculated as follows:

$$Q_{NP} = \frac{2\,k_N k_P}{\Delta s(k_N + k_p)}\ (\Delta s)(T_N - T_P) \tag{4}$$

$$Q_{WP} = \frac{2k_W k_P}{\Delta s(k_W + k_P)}\ (\Delta s)(T_W - T_P) \tag{5}$$

$$Q_{SP} = \frac{2k_S k_P}{\Delta s(k_S + k_P)}(\Delta s)(T_S - T_P) \tag{6}$$

$$Q_{EP} = \frac{2k_E k_P}{\Delta s(k_E + k_P)}\ (\Delta s)(T_E - T_P) \tag{7}$$

Note that the area perpendicular to the direction of heat flow is taken per unit depth, that is, $A = \Delta s \cdot 1$.

The sum of the heat flows into a node is responsible for the temperature change of the node:

$$Q_{NP} + Q_{WP} + Q_{SP} + Q_{EP} = V_P \cdot \frac{1}{\Delta t}\ (\rho' C_P' T_P' - \rho C_P T_P) \tag{8}$$

where V_P is the control volume for unit depth ($V_P = (\Delta s)^2$ in this case) and the primed variables represent time $t + \Delta t$.

Now substitute eq 4–7 into eq 8 to get the finite difference form of eq 1 for a node not on the boundary of the grid:

$$k_{NP}(T_N - T_P) + k_{WP}(T_W - T_P) + k_{SP}(T_s - T_P) + k_E(T_E - T_P) = \frac{(\Delta s)^2}{\Delta t}\ (\rho' C_P' T_P' - \rho\, C_P T_P). \tag{9}$$

Since Δs is uniform for interior nodes of the grid, substitutions like the following have been made for simplicity's sake:

$$k_{NP} = \frac{2k_N k_P}{k_N + k_P}\,. \tag{10}$$

Equation 8 describes the temperature change for one node P over one time step. The values ρ, C_P and k are held constant for each time step, but may be changed between steps. It is necessary to apply eq 8 (or the appropriate boundary equation) to each node of the grid to solve for the temperature distribution for one time step. The process is repeated until the desired number of time steps have been completed. Let us now investigate several methods used in the solution of the set of equations.

The first and probably the easiest method to program is the explicit method, where eq 8 is solved for T_P' (which is the value of T_P for time $t + \Delta t$). Note that all of the other temperature variables represent time t, so that the equation may be solved explicitly. An initial temperature is assigned to each node in the grid; the appropriate equation is solved for each node to determine the temperature resulting from the first time step. The resultant temperature distribution is then used in calculations for the second time step. This process continues until the desired number of time steps have been calculated. The main problem with this method is that a stability criterion must be met to produce a reliable answer. For a two-dimensional homogeneous grid, the criterion is

$$\frac{(\Delta s)^2\ \rho C_p}{k \Delta t} > 4.$$

Note that a balance must be struck between the internodal distance (Δs) and the time step (Δt). In problems involving steep temperature gradients, a small internodal spacing must be used; consequently, the time step must be very small. The combination of many nodes and time steps makes the solution costly in computer time.

A second method used in solving finite difference equations is the implicit method. Here T'_P is taken as the value of T_P for time t, and all of the other variables represent time $t + \Delta t$; hence, all temperature variables are unknown except T'_P. An appropriate equation must be written for each node in the grid, and the resulting set of equations is solved simultaneously for each time step. For the two-dimensional case of an x by y grid, there will be xy equations in xy unknowns. The matrix of coefficients is banded (bandwidth = $2x + 1$), but in general there is no symmetry about the main diagonal. Large matrices often occur and they are usually solved by an iterative method. The implicit method has the advantage of being unconditionally stable, but it may require a fair number of iterations for adequate convergence in the matrix solution.

The finite difference method chosen for this computer program is known as the alternating direction implicit method (Peaceman and Rachford 1955, Carnahan et al. 1969). It is unconditionally stable like the fully implicit method, yet it requires only the solution of a tridiagonal matrix, thus being efficient in computer time and computer core storage. (The matrix-solving algorithm is discussed in the *TRIDIG, The Matrix Solver* section). Essentially, the method requires the solution of two different equations for each time step. The first equation is implicit only in the horizontal direction. The results from the first step are then used to solve the second equation, which is implicit only in the vertical direction.

For example, apply the alternating direction considerations to a node in the interior of the grid. Equation 9, which describes the heat balance for an interior node, is repeated here for convenience:

$$k_{NP}(T_N - T_P) + k_{WP}(T_W - T_P) + k_{SP}(T_S - T_P) + k_{EP}(T_E - T_P) = \frac{(\Delta s)^2 \rho' C'_P}{\Delta t} T'_P - \frac{(\Delta s)^2 \rho C_P}{\Delta t} T_P .$$

Each node in the grid is given an initial temperature. Then, for the first pass, let the above equation be implicit in the horizontal direction. The time step for the pass will be $\Delta t/2$, and the above equation may be written as follows:

$$k_{WP}T'_W + k_{EP}T'_E - \left[k_{WP} + k_{EP} + \frac{2(\Delta s)^2 \rho' C'_P}{\Delta t}\right] T'_P = -k_{NP}T_N - k_{SP}T_S + \left[k_{NP} + k_{SP} - \frac{2(\Delta s)^2 \rho C_P}{\Delta t}\right] T_P . \tag{11}$$

The primed variables represent the value of those variables at time $t + \Delta t/2$. The right-hand side of the equation is known; T'_W, T'_P and T'_E are to be determined. Equation 11 (or the appropriate boundary equation) is applied to each node of a row of the grid, creating a tridiagonal matrix of coefficients. Such a system of equations is quickly solved for each row of the grid. The resulting temperature distribution represents $t + \Delta t/2$, the end of the first pass.

The process is similar for the second pass, except that now eq 9 is rewritten to be implicit in the vertical direction:

$$k_{NP}T'_N + k_{SP}T'_S - \left[k_{NP} + k_{SP} + \frac{2(\Delta s)^2 \rho' C'_P}{\Delta t}\right] T'_P = - k_{WP}T_W - k_{EP}T_E + \left[k_{WP} + k_{EP} - \frac{2(\Delta s)^2 \rho C_P}{\Delta t}\right] T_P . \tag{12}$$

The primed variables represent time $t + \Delta t$, and the others represent time $t + \Delta t/2$. The tridiagonal matrix is formed and solved for each column of the grid; the resultant temperatures represent the temperature distribution in the grid after $t + \Delta t$, one time step. The entire process is repeated until the desired number of steps have been calculated.

The reader may consult Carnahan et al. (1969), Holman (1972), Croft and Lilley (1977) and Mitchell and Griffiths (1980) for more information on finite difference methods.

Boundary conditions

The following boundary conditions will be derived from heat balance considerations. In this paper each boundary equation will not be expressed in the form needed for use in the alternating direction procedure, but the reader should be aware that each equation was put into that form for use in the computer programs ADI and ADIPC.

Sides of the grid

Consider a node, P, on the right-hand grid boundary. The control volume associated with the node is the area enclosed by dotted lines around it, as depicted in Figure 3.

The sum of the heat flowing through the boundaries of P's control volume is responsible for the temperature change of the node, that is,

$$Q_{NP} + Q_{WP} + Q_{SP} + Q_{EP} = \frac{\Delta x \, \Delta y \, \rho \, C_P \, \Delta T_P}{2\Delta t} \tag{13}$$

where Q_{NP} = the heat flow from node N to (or from) node P

$\frac{\Delta x}{2} \cdot \Delta y$ = the volume (for unit depth) of node P

ρ = the density of the material in node P

C_P = the specific heat

T = temperature

t = time.

The flow of heat from node N to node P is given by

$$Q_{NP} = \frac{1}{2} k_{NP} (T_N - T_P) \tag{14}$$

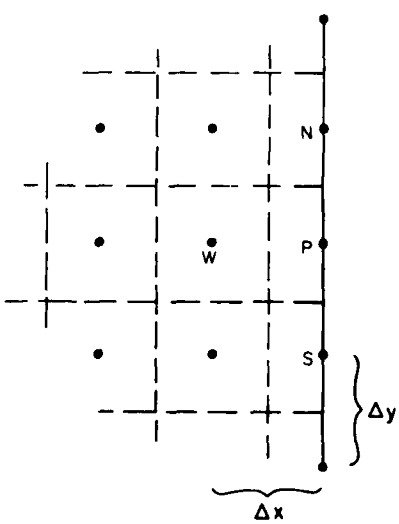


Figure 3. A node on a right-hand boundary.

where k_{NP} is the effective conductivity between nodes N and P, and for $\Delta x = \Delta y = \Delta s$,

$$k_{NP} = \frac{2k_N k_P}{(k_N + k_P)}$$

Similarly, the heat flow from nodes W and S to node P may be given as

$$Q_{WP} = k_{WP}(T_W - T_P) \tag{15}$$

$$Q_{SP} = \frac{1}{2} k_{SP}(T_S - T_P) . \tag{16}$$

Now consider several cases describing the heat flow across the boundary.

Constant flux boundary. For a boundary subject to a constant heat flux, the condition

$$\left.\frac{\partial T}{\partial x}\right|_{x=P} = \text{constant}$$

exists at the boundary. For node P, illustrated in Figure 3, the heat flow across its east side is given by

$$Q_{EP} = \phi \cdot \Delta y \tag{17}$$

where ϕ is the heat flux per unit area crossing the boundary. The equation for a node on the right-hand side of the grid with constant flux is obtained by combining eq 13-17 and allowing $\Delta x = \Delta y = \Delta s$:

$$\frac{1}{2} k_{NP}(T_N - T_P) + k_{WP}(T_W - T_P) + \frac{1}{2} k_{SP}(T_S - T_P) + \phi\Delta s = \frac{(\Delta s)^2}{2\Delta t}[\rho' C_P' T_P' - \rho C_P T_P] . \tag{18}$$

The indices may be suitably rearranged for constant flux boundaries on other sides of the grid.

Note that for a boundary which is insulated or on a line of symmetry the zero heat flux condition holds, and $\phi = 0$.

Convection boundary. For the node illustrated in Figure 3, exposed to convection on the right-hand side, the heat flow from the convective medium to node P is given by

$$Q_{EP} = h_E \Delta y (T_E - T_P) \tag{19}$$

where h_E is the coefficient of convective heat transfer and T_E is the temperature outside the grid. Combining equations 13, 14, 15, 16 and 19, we arrive at the equation for a node on the convective right-hand side boundary, for $\Delta x = \Delta y = \Delta s$:

$$\frac{1}{2} k_{NP}(T_N - T_P) + k_{WP}(T_W - T_P) + \frac{1}{2} k_{SP}(T_S - T_P) + h\Delta s(T_E - T_P) =$$

$$\frac{(\Delta s)^2}{2\Delta t}[\rho' C_P' T_P' - \rho C_P T_P] . \tag{20}$$

Specified temperature boundary. For a node of specified temperature on a boundary, corner or inside the grid, apply the equation $T_P = C$, where C is the temperature of the node at time t.

Semi-infinite boundary. This condition represents a continuous, uniform material extending in one direction, with a known temperature a large distance away. It is approximated here by use

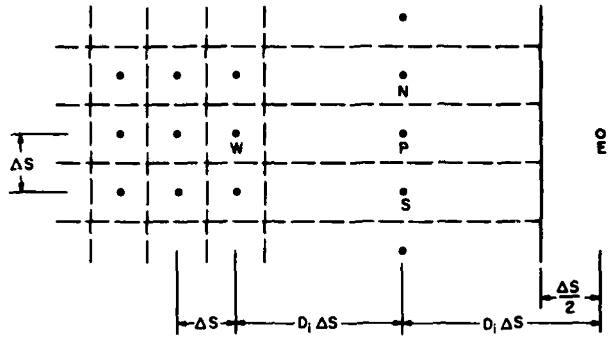


Figure 4. A node on a semi-infinite boundary.

of a large internodal distance between the last two nodes of a row of the grid. Consider the situation illustrated in Figure 4, for the right-hand side of the grid. Node E is not actually a part of the grid, but is the location of known temperature T_E outside the grid.

The distance between the last two nodes and also between the boundary node and the location of the known temperature T_E is $D_i \Delta s$. Note that D_i represents a multiple of Δs. The heat flow from nodes N, W, S and E to node P may be calculated as follows

$$Q_{NP} = k_{NP}(2D_i - 1)(T_N - T_P) \tag{21}$$

$$Q_{WP} = k_{WP}(T_W - T_P) \tag{22}$$

$$Q_{SP} = k_{SP}(2D_i - 1)(T_S - T_P) \tag{23}$$

$$Q_{EP} = k_{EP}(T_E - T_P)\,. \tag{24}$$

The effective conductivities are of the same form as eq 10, except for k_{WP} and k_{EP}

$$k_{WP} = \frac{2k_W k_P}{[k_P + (2D_i - 1)k_W]}\,.$$

Node E is assumed to be the same material as node P, thus $k_{EP} = k_P$. The sum of the heat flows accounts for the temperature change of the node, thus the equation for a node on the right-hand semi-infinite boundary is

$$k_{NP}(2D_i - 1)(T_N - T_P) + k_{WP}(T_W - T_P) + k_{SP}(2D_i - 1)(T_S - T_P) +$$

$$k_{EP}(T_E - T_P) = \frac{(2D_i - 1)(\Delta s)^2}{\Delta t}\,(\rho' C_P' T_P' - \rho\, C_P\, T_P) \tag{25}$$

where $(2D_i - 1)(\Delta s)^2$ is the control volume for the node per unit depth.

In any finite difference formulation, the accuracy of the solution increases as the area represented by a node decreases. Therefore, when using the semi-infinite boundary formulation, the user should specify the smallest D_i acceptable. The semi-infinite condition should only be used in regions where the temperature gradient is small and precise knowledge of the temperature distribution is not critical.

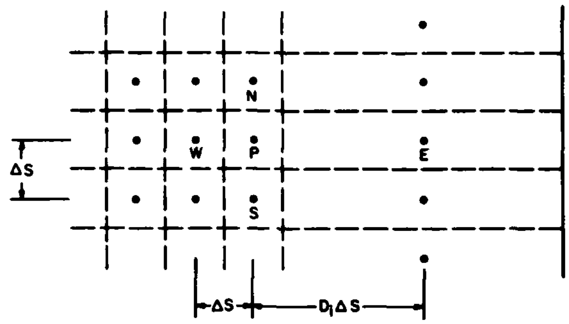


Figure 5. An interior node adjacent to the semi-infinite boundary.

This boundary conditions also requires a special heat balance for the node adjacent to the semi-infinite node. Consider again the situation for a right-hand boundary, as illustrated in Figure 5. The appropriate heat flow equations are

$$Q_{NP} = k_{NP}\,(T_N - T_P) \tag{26}$$

$$Q_{WP} = k_{WP}\,(T_W - T_P) \tag{27}$$

$$Q_{SP} = k_{SP}\,(T_S - T_P) \tag{28}$$

where k_{NP}, k_{WP}, k_{SP} are of the same form as eq 10, and

$$Q_{EP} = k_{EP}\,(T_N - T_P) \tag{29}$$

where $k_{EP} = \dfrac{2k_E k_P}{[k_E + (2D_i - 1)\,k_P]}$.

The equation for an interior node adjacent to a right-hand semi-infinite boundary node is

$$k_{NP}\,(T_N - T_P) + k_{WP}\,(T_W - T_P) + k_{SP}\,(T_S - T_P) + k_{EP}\,(T_E - T_P) =$$

$$\frac{(\Delta s)^2}{\Delta t}\;[\rho'\,C_P'\,T_P' - \rho\,C_P\,T_P]\;. \tag{30}$$

Corner of the grid

The corners of the grid require special heat balances, depending on the conditions on each edge of the grid. Consider the upper right-hand corner of the grid, shown in Figure 6. T_N and T_E are not a part of the grid, but are temperatures outside the grid. The heat flow from the two nodes adjacent to node P may be given as

$$Q_{WP} = \frac{1}{2}\,k_{WP}\,(T_W - T_P) \tag{31}$$

$$Q_{SP} = \frac{1}{2}\,k_{SP}\,(T_S - T_P) \tag{32}$$

where k_{WP} and k_{SP} follow the form given in eq 10. Q_{NP} and Q_{EP} are dependent upon the

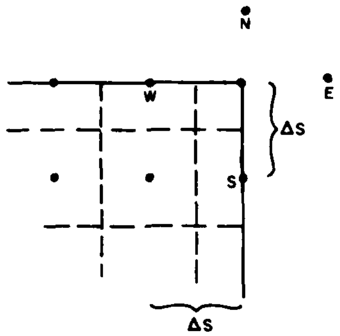


Figure 6. A node on a corner of the grid.

particular boundary conditions and will be presented shortly. The corner equations will follow the form

$$Q_{NP} + Q_{WP} + Q_{SP} + Q_{EP} = \frac{(\Delta s)^2}{4} \rho C_P \frac{\Delta T}{\Delta t} \tag{33}$$

where $\frac{(\Delta s)^2}{4}$ is the volume represented by node P for unit depth.

Constant flux on both sides. Let ϕ_N be the heat flux per unit area crossing the north side of the corner shown in Figure 6, and ϕ_E be that crossing the east side of the corner. Then the heat flows across the two sides are given by

$$Q_{NP} = \phi_N \cdot \frac{\Delta s}{2} \tag{34}$$

$$Q_{EP} = \phi_E \cdot \frac{\Delta s}{2} \,. \tag{35}$$

The equation for the corner is

$$\frac{1}{2} k_{WP} (T_W - T_P) + \frac{1}{2} k_{SP} (T_S - T_P) + \frac{1}{2} \Delta s (\phi_N + \phi_E) = \frac{(\Delta s)^2}{4\Delta t} [\rho' C_P' T_P' - \rho C_P T_P] \,. \tag{36}$$

As in the case of the side of the grid, if a side of the corner is insulated or on a line of symmetry, $\phi = 0$ for that side.

Convection on both sides. For a corner subject to convection on both of its sides, the heat flow from each of the two sides may be given as

$$Q_{NP} = h_N \frac{\Delta s}{2} (T_N - T_P) \tag{37}$$

$$Q_{EP} = h_E \frac{\Delta s}{2} (T_E - T_P) \tag{38}$$

where h_N is the coefficient of convective heat transfer on the north side and h_E is that on the east side. The equation for node P is

$$\frac{1}{2} k_{WP} (T_W - T_P) + \frac{1}{2} k_{SP} (T_S - T_P) + \frac{1}{2} h_N \Delta s (T_N - T_P) + \frac{1}{2} h_E \Delta s (T_E - T_P) = \frac{(\Delta s)^2}{4\Delta t} [\rho' C_P' T_P' - \rho C_P T_P] \,. \tag{39}$$

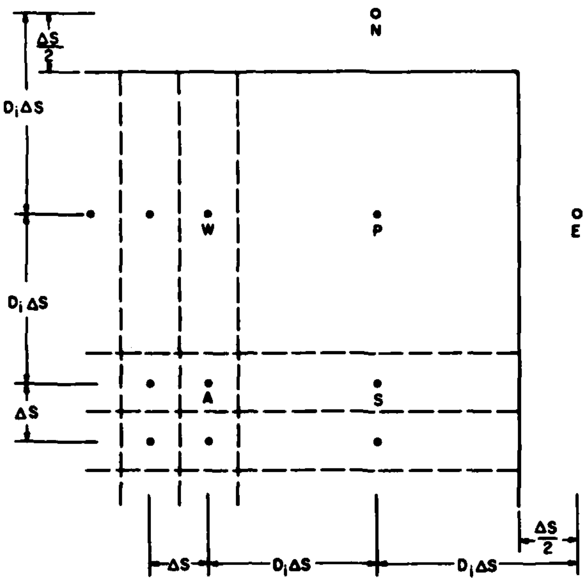


Figure 7. A semi-infinite corner.

Semi-infinite on both sides. For a corner with semi-infinite conditions on both sides, an irregular finite difference grid is employed once more. Consider again the upper right-hand corner, now illustrated in Figure 7. T_N and T_E are known temperatures outside the grid. Heat flows are given by

$$Q_{NP} = k_{NP}(2D_i - 1)(T_N - T_P) \tag{40}$$

$$Q_{WP} = k_{WP}(2D_i - 1)(T_W - T_P) \tag{41}$$

$$Q_{SP} = k_{SP}(2D_i - 1)(T_S - T_P) \tag{42}$$

$$Q_{EP} = k_{EP}(2D_i - 1)(T_E - T_P)\,. \tag{43}$$

Again,

$$k_{NP} = k_{EP} = k_P;\, k_{WP} = \frac{2k_W k_P}{k_P + (2D_i - 1)\,k_W} \text{ and } k_{SP} = \frac{2k_S k_P}{k_P + (2D_i - 1)\,k_S}\,.$$

Then the equation for a semi-infinite corner node follows:

$$k_{NP}(2D_i - 1)(T_N - T_P) + k_{WP}(2D_i - 1)(T_W - T_P) + k_{SP}(2D_i - 1)(T_S - T_P) +$$

$$k_{EP}(2D_i - 1)(T_E - T_P) = \frac{[(2D_i - 1)(\Delta s)]^2}{\Delta t}\;[\rho' C_P' T_P' - \rho C_P T_P]\,. \tag{44}$$

Because the nodes are of irregular size, special consideration must be given also to node temperatures T_W, T_S and T_A in Figure 7.

Consider first node A. This is similar to the node P illustrated in Figure 5, except that the distance to node N is now $D_i \cdot \Delta s$. This results in our changing k_{NP} to

$$k_{NP} = \frac{2k_N k_P}{k_N + (2D_i - 1)\,k_P}\,.$$

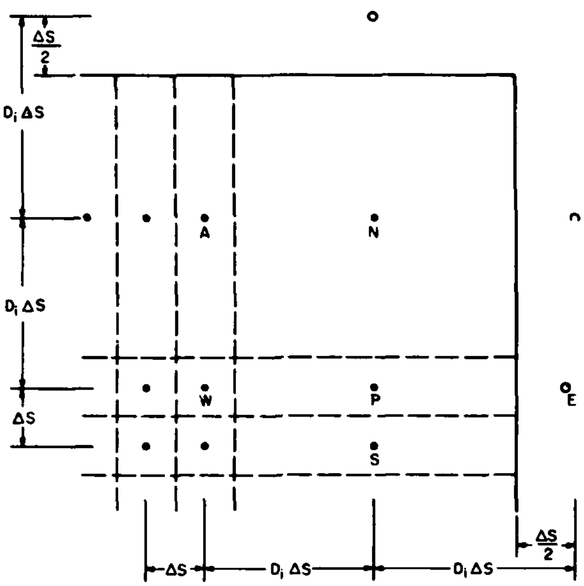


Figure 8. A semi-infinite side node adjacent to a semi-infinite corner.

Except for this change in k_{NP}, the equation for the square interior node inside an upper right-hand corner with semi-infinite conditions on both sides is the same as eq 29.

Now consider the semi-infinite nodes adjacent to the semi-infinite corner node. These are node temperatures T_W and T_S in Figure 7. The figure is presented again, this time as Figure 8, with the nodes relabeled. The type of node under consideration is represented by nodes A and P in Figure 8. The following equations reflect the heat flow to node P:

$$Q_{NP} = k_{NP}(2D_i - 1)(T_N - T_P) \tag{45}$$

$$Q_{WP} = k_{WP}(T_W - T_P) \tag{46}$$

$$Q_{SP} = k_{SP}(2D_i - 1)(T_S - T_P) \tag{47}$$

$$Q_{EP} = k_{EP}(T_S - T_P)\,. \tag{48}$$

Here,

$$k_{NP} = \frac{2k_N k_P}{k_N + (2D_i - 1)k_P} \quad \text{and } k_{WP} = \frac{2k_W k_P}{k_P + (2D_i - 1)k_W}\,.$$

The equation for this node is

$$k_{NP}(2D_i - 1)(T_N - T_P) + k_{WP}(T_W - T_P) + k_{SP}(2D_i - 1)(T_S - T_P) +$$

$$k_{EP}(T_E - T_P) = \frac{(2D_i - 1)(\Delta s)^2}{\Delta t}\,[\rho' C_P' T_P' - \rho C_P T_P]\,. \tag{49}$$

Constant flux, convective corner. For the corner illustrated in Figure 6, allow convection to occur across the north side of node P and constant flux across the east side

$$Q_{NP} = h_N \frac{\Delta s}{2} (T_N - T_P) \tag{50}$$

Q_{EP} is given by eq 35, and Q_{WP} and Q_{SP} are given by eq 31 and 32, respectively. The resulting equation for the corner is:

$$\frac{1}{2} h_N \Delta s (T_N - T_P) + \frac{1}{2} k_{WP} (T_W - T_P) + \frac{1}{2} k_{SP} (T_S - T_P) + \frac{1}{2} \phi_E \Delta s =$$

$$\frac{(\Delta s)^2}{4\Delta t} [\rho' C_P' T_P' - \rho C_P T_P] \; . \tag{51}$$

Constant flux, semi-infinite corner. The corner illustrated in Figure 9 has a semi-infinite boundary on the right side. Let there be a constant flux per unit area, ϕ_N, across the north side of node P, then

$$Q_{NP} = \phi_N (2D_i - 1) \Delta s \tag{52}$$

$$Q_{WP} = \frac{1}{2} k_{WP} (T_W - T_P) \tag{53}$$

$$Q_{SP} = k_{SP} (2D_i - 1)(T_S - T_P) \tag{54}$$

$$Q_{EP} = \frac{1}{2} k_{EP} (T_E - T_P)$$

$$k_{WP} = \frac{2k_W k_P}{k_P + (2D_i - 1) k_W}$$

$$k_{SP} = \frac{2k_S k_P}{k_S + k_P}$$

$$k_{EP} = k_P \; . \tag{55}$$

The equation for the corner is as follows:

$$\phi_N (2D_i - 1) \Delta s + \frac{k_{WP}}{2} (T_W - T_P) + k_{SP} (2D_i - 1)(T_S - T_P) + \frac{k_{EP}}{2} (T_E - T_P) =$$

$$\frac{(2D_i - 1)(\Delta s)^2}{2\Delta t} [\rho' C_P' T_P' - \rho C_P T_P] \; . \tag{56}$$

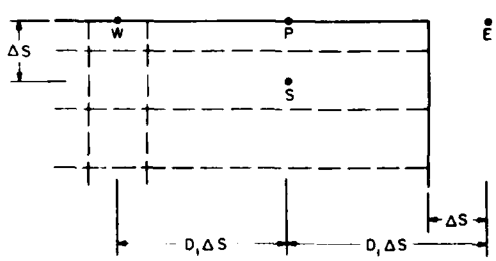


Figure 9. A node on a semi-infinite corner.

Again, the node adjacent to the semi-infinite node must have a special equation. For node W in Figure 9, with constant flux ϕ_N across the top, eq 57 applies, with referance to the node under consideration as node P:

$$\phi_N \Delta s + \frac{1}{2} k_{WP} (T_W - T_P) + k_{SP} (T_S - T_P) + \frac{1}{2} k_{EP} (T_E - T_P) =$$

$$\frac{(\Delta s)^2}{2\Delta t} [\rho' C'_P T'_P - \rho C_P T_P] . \quad (57)$$

Semi-infinite, convective corner. For the corner in Figure 9, allow convection across the north side, and allow the same semi-infinite boundary on the east side,

$$Q_{NP} = h_N (2D_i - 1)(T_N - T_P) . \quad (58)$$

Q_{WP}, Q_{SP} and Q_{EP} appear in eq 53, 54 and 55, respectively. Thus the equation for the configuration is

$$h_N (2D_i - 1)(T_N - T_P) + \frac{k_{WP}}{2} (T_W - T_P) + k_{SP} (2D_i - 1)(T_S - T_P) + \frac{k_{EP}}{2} (T_E - T_F) =$$

$$\frac{(2D_i - 1)(\Delta s)^2}{2\Delta t} [\rho' C'_P T'_P - \rho C_P T_P] . \quad (59)$$

The appropriate equation for node W in Figure 9, subject to convection across the top, follows. Again, refer to this node as node P:

$$h_N \Delta s (T_N - T_P) + \frac{1}{2} k_{WP} (T_W - T_P) + k_{SP} (T_S - T_P) + \frac{k_{WP}}{2} (T_W - T_P) =$$

$$\frac{(\Delta s)^2}{2\Delta t} [\rho' C'_P T'_P - \rho C_P T_P] . \quad (60)$$

PHASE CHANGE

The methods discussed in this report so far apply to both ADIPC and ADI, the programs developed for heat conduction with and without phase change. The ideas on phase change presented in this section will apply only to ADIPC. The computer program solves the heat conduction equation only; possible effects of moisture migration through the medium (and unfrozen water content in a frozen soil, for example) are neglected. Although the examples used involve freezing, ADIPC permits either freezing or thawing at any node of the grid.

During the freezing process, the temperature of a substance that is initially above its fusion temperature decreases as heat is removed until the substance reaches its fusion temperature. The continuing removal of heat produces no change in temperature until the energy equivalent to the latent heat of the substance is removed. The substance has then changed state to become frozen; any further removal of heat again results in a decrease of temperature below the temperature of fusion.

ADIPC models freezing or thawing by defining an apparent specific heat during phase change that accounts for the entire enthalpy change that takes place, including the enthalpy in the latent heat of fusion. To do this, it must be assumed that phase change takes place over a finite temperature range, ΔT, around the fusion temperature. Allow T_f to denote the fusion temperature, C_{P_A}

and C_{P_B} to be the specific heat of the substance above and below freezing, respectively, and let $C_P{}^*$ be the apparent specific heat defined for phase change. For a temperature change in a substance not involving a change of state, the enthalpy change ΔH is given as

$$\Delta H = \int C_P \, dT .$$

Now the temperature range for phase change will be $T_f \pm \Delta T/2$, and the apparent specific heat accounts for the entire enthalpy change as follows, where H_L represents the latent heat of fusion:

$$\int_{T_f - \frac{\Delta T}{2}}^{T_f + \frac{\Delta T}{2}} C_P^* \, dT = \int_{T_f - \frac{\Delta T}{2}}^{T_f} C_{P_B} \, dT + \int_{T_f}^{T_f + \frac{\Delta T}{2}} C_{P_A} \, dT + H_L . \tag{61}$$

ΔT is chosen small enough so that each specific heat may be assumed constant over the integral,

$$C_P^* \Delta T = C_{P_B} \frac{\Delta T}{2} + C_{P_A} \frac{\Delta T}{2} + H_L$$

$$C_P^* = \frac{1}{2} (C_{P_B} + C_{P_A}) + \frac{H_L}{\Delta T} . \tag{62}$$

Equation 62 defines the apparent heat capacity as used in ADIPC. The apparent heat capacity method is further discussed in Bonacina and Comini (1973).

Phase change is implemented in ADIPC as follows. At the beginning of each time step, each node of the grid is examined. If its temperature lies within the range $T_f \pm \Delta T/2$, the specific heat for that node is defined by eq 62. The conduction equations are set up and solved as usual. At the end of the complete time step, the newly calculated temperature of each node is compared with the temperature of that node at the beginning of the time step. If the temperature skipped from below $T_f - \Delta T/2$ to above $T_f + \Delta T/2$, or vise versa, then the phase change front skipped that node, and the temperature of the node is reassigned as follows. If the temperature of the node skipped from the frozen to the unfrozen domain,

$$T = \left(T_f - \frac{\Delta T}{2}\right) + \frac{C_{P_B}}{C_P^*} \left[T' - \left(T_f - \frac{\Delta T}{2}\right)\right] \tag{63}$$

where T' is the calculated nodal temperature before its reassignment. If the node skipped from an unfrozen state to the frozen state, its temperature is reassigned as follows:

$$T = \left(T_f + \frac{\Delta T}{2}\right) + \frac{C_{P_A}}{C_P^*} \left[T' - \left(T_f + \frac{\Delta T}{2}\right)\right] . \tag{64}$$

In this way, the program assures that the phase change front does not skip a node.

A drawback with the apparent heat capacity method is that it is not designed to follow the exact location of the phase change front for each time step; it is designed to calculate nodal temperature. The location of the T_f isotherm may be found by interpolation, but it approximates the location of the front in a step-like pattern because of the discretization of a continuous space, as will be illustrated in the program verifications. This behavior is minimized by use of a small internodal spacing. The method is flexible enough to deal with phase change in two-dimensional space when there may be several locations of phase change fronts; for this flexibility the method was chosen.

COMPUTER PROGRAM

In using finite differences, we replace differentials and derivatives with differences. It makes sense, then, that the accuracy of the solution increases as these differences become smaller. Specifically, the accuracy improves as the nodal spacing and time step, Δs and Δt, are made small. There is no set criterion for just how small they must be, usually this is discovered through trial and error.

The grid for these programs may be any size, so far as the outside dimensions are concerned, but, except for the semi-infinite boundary conditions, each node in the grid represents a square area of unit depth. RAY (I, J, K) is the array that represents the grid. I and J are spatial variables in Cartesian coordinates; I is incremented vertically and J horizontally. K has three values. RAY (I, J, 1) represents the temperature distribution in the grid, i.e., a temperature is stored at each I, J location. RAY (I, J, 2) records the nodal location type. Examples of location types include a node on a constant flux boundary, variable interior node, specified temperature corner node, etc. Each location type is assigned a number, and one such number is stored for each I, J location under RAY (I, J, 2). This information is used to assign the correct form of the heat conduction equation to each node. Each material type in the problem is given a number; these numbers are stored in RAY (I, J, 3) and may be used to assign a conductivity, density and specific heat for each node.

The programs are made up of four parts: 1) a data-gathering subroutine, 2) the main program, 3) a subroutine to solve the tridiagonal matrix and 4) a subroutine to locate the user-specified isotherms at times specified by the user. Parts 3 and 4 are identical for both ADI and ADIPC, and part 1 is similar for each. Therefore, parts 1, 3 and 4 will be discussed only once but are included in each of the two programs.

ADDATA, the data subroutine

The necessary data for the program are handled mainly through subroutine ADDATA. Variables used are defined in the comment statements at the beginning of the subroutine. The user has simply to edit the subroutine, following directions in the comment statements in the subroutine to initialize variables and arrays. When the subroutine is run, the data are put into a formatted data file ADIDAT; the user does not have to worry about formats.

TRIDIG, the matrix solver

This subroutine solves the tridiagonal matrix formed in the main program. The matrix is formed by the application of eq 11 or 12 (or a boundary condition counterpart) to each node of the grid. The resultant system of equations is illustrated in matrix form in Figure 10. The matrix of coefficients is an n by n matrix; all entries are zero except those on the three center diagonals, hence the term tridiagonal matrix. The vector on the right contains the elements of the right-hand side of each equation. To conserve computer storage space, only the three diagonals are stored, as vectors. Thus the effective storage space required is reduced from n by $n + 1$ to n by 4.

The solution algorithm is commonly used (e.g. Gerald 1980). Each element of the lower diagonal may be eliminated by subtracting the appropriate multiple of the $(i - 1)$ row from the ith row. The values of b_i and d_i, after elimination of a_i, are

$$b_i = b_i - \left(\frac{a_i}{b_{i-1}}\right) c_{i-1} \tag{65}$$

$$d_i = d_i - \left(\frac{a_i}{b_{i-1}}\right) d_{i-1} \quad \text{for } i = 1, 2, 3 \ldots n . \tag{66}$$

$$\begin{bmatrix} b_1 & c_1 & & & & \\ a_2 & b_2 & c_2 & & 0 & \\ & a_3 & b_3 & c_3 & & \\ & & \ddots & \ddots & \ddots & \\ & 0 & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & & a_n & b_n \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ \vdots \\ T_{n-1} \\ T_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix}$$

Figure 10. Tridiagonal matrix equation.

After replacing the element of vectors b and d by the new values, we perform a back substitution as follows:

$$d_n = \frac{d_n}{b_n} \tag{67}$$

$$d_i = \frac{d_i - c_i d_{i+1}}{b_i}\,, \quad \text{for } i = n-1, n-2, \ldots 1\,. \tag{68}$$

The elements of the solution vector replace vector d.

ISOTHM, the isotherm finder

Subroutine ISOTHM examines the temperature distribution in the grid and performs a linear interpolation between adjacent nodes to produce the Cartesian coordinates of the locations of the user-specified isotherms. The coordinates are listed in file POINTS in two columns, one for the horizontal coordinate and one for the vertical coordinate. POINTS may be used as a data file for a plotting routine. Subroutine ISOTHM may be called at the completion of any number of time steps; the frequency is specified by the user in ADDATA.

ADI, main program

ADI first interactively asks the user if ADDATA should be called. The subroutine should be called if ADIDAT does not already contain the formatted data. If the user so desires ADDATA is called. If not, ADI reads the data from ADIDAT.

For each time step, the thermal properties such as conductivity, density and specific heat are updated, if appropriate, and the resultant conductivities between the nodes are figured. Each node is assigned the coefficients for the appropriate form of the conduction equation, forming the tridiagonal matrix (Fig. 10), and subroutine TRIDIG is called to solve the matrix. The resultant temperature distribution is then used in the second half-time step, and TRIDIG is called once again. These temperatures then represent the distribution for one complete time step. If appropriate, subroutine ISOTHM is then called to locate the user-specified isotherms. This procedure is repeated for the desired number of time steps.

The initial data and boundary conditions are printed into file ADIOUT; the temperature distributions for specified time steps are printed into file ADITMP, and finally, a new data file, ADNDAT, is created for the final temperature distribution. This file is in the same format as the input file ADIDAT; ADNDAT may be used as a starting point if the user wishes to run the model for more time.

ADIPC, main program

ADIPC is similar to ADI, but differs in that phase change considerations are implemented. At the beginning of each time step, if the temperature of a node lies in the phase change temperature range, the program assigns the apparent specific heat to that node, as previously discussed. Otherwise, the specific heat, thermal conductivity and density for each node are updated according to the user's specifications.

At the end of the time step, the temperature of each node is compared to its temperature at the end of the previous time step. If the temperature skipped over the phase change temperature range, it is reassigned as discussed in the *Phase Change* section. Then, if appropriate, subroutine ISOTHM is called.

The initial and boundary conditions are printed in readable form in file ADPOUT; the temperature distributions for specified time steps are printed into file ADPTMP, and a new data file with the final temperature distributions is printed into file ADPNDT.

VERIFICATION OF ADI

Comparison of ADI with analytical results

Semi-infinite corner

The results of ADI will first be compared to the problem of a semi-infinite corner, as illustrated in Figure 11.

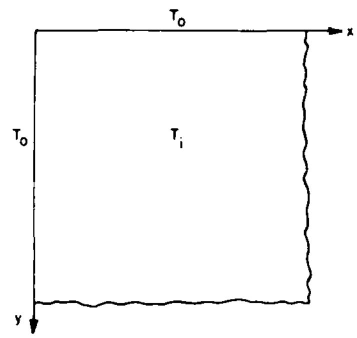


Figure 11. Semi-infinite corner problem.

T_i is the uniform initial temperature; at time $t = 0$ the temperature of the two edges, given by $x = 0$ and $y = 0$, is changed to T_0. The solution is well documented (Carslaw and Jaeger 1959, and Holman 1972) and is found by using a product solution for two one-dimensional problems. The solution is given by

$$\frac{T - T_0}{T_i - T_0} = \operatorname{erf} \frac{x}{2\sqrt{\alpha t}} \operatorname{erf} \frac{y}{2\sqrt{\alpha t}} \tag{69}$$

where α is the thermal diffusivity, and

$$\operatorname{erf} \frac{x}{2\sqrt{\alpha t}} = \frac{2}{\sqrt{\pi}} \int_0^{x/(2\sqrt{\alpha t})} e^{-\eta^2} \, d\eta .$$

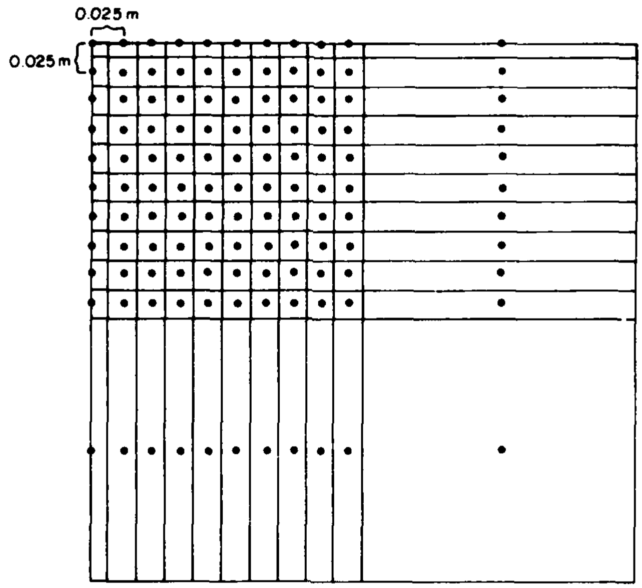


Figure 12. Finite difference grid for the semi-infinite corner problem.

For a comparison run, the following values were used:

$T_i = 20°C\ (68°F)$

$T_0 = 40°C\ (104°F)$

$\alpha = 0.0025\ m^2/hr$

The finite difference grid appears as shown in Figure 12. The boundary condition assigned to the top and left side was that of constant temperature equal to 40°C. The right side and bottom of the grid were assigned a semi-infinite boundary condition, with $T_S = T_E = 20°C$ and $D_i = 50$ (see the *Sides of the Grid* section). The nodes not situated on a boundary were assigned an initial temperature of 20°C. A time step of 0.25 hr was used, and the internodal spacing was 0.025 m.

The locations of the 35°, 30° and 25°C isotherms are plotted for the results of ADI and for the analytical solution for several time steps in Figure 13. Excellent agreement is found between the two solutions for regions not adjacent to the semi-infinite boundary. As previously stated, the semi-infinite condition is an approximation. Also, the large internodal distance used for the semi-infinite boundary decreases the accuracy of the solution in that region.

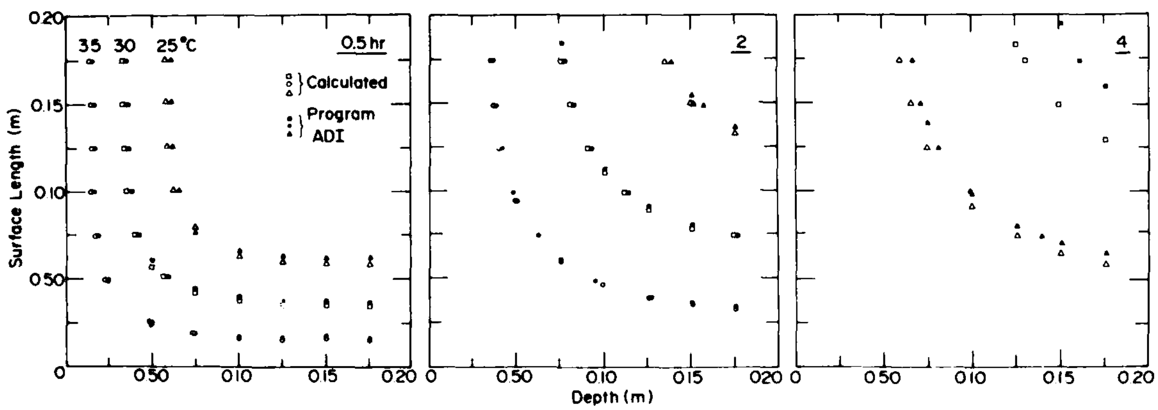


Figure 13. Locations of 35°, 30° and 25° C isotherms from ADI and from the analytical solution for the semi-infinite corner problem.

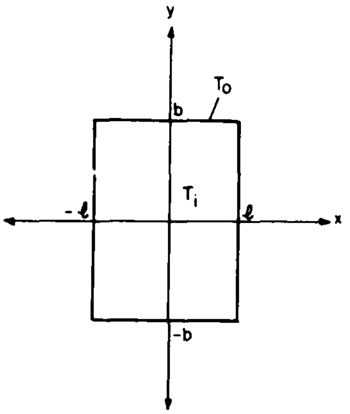


Figure 14. Finite rectangle problem.

Finite rectangle

In order to provide a two-dimensional comparison that doesn't involve the use of semi-infinite boundaries, and also to demonstrate internodal spacing effects, consider the problem of a rectangle of uniform initial temperature T_i that is subject to a step change in the temperatures of all of the edges to T_0 at time zero. The problem is illustrated in Figure 14. The solution of this problem is again obtainable from the product solution of two one-dimensional problems. Carslaw and Jaeger (1959) provide the one dimensional solution:

$$\frac{T(x,t)-T_0}{T_i-T_0} = \frac{4}{\pi}\sum_{n=0}^{\infty}\frac{(-1)^n}{2n+1}\exp\frac{-\alpha(2n+1)^2\pi^2 t}{4\ell^2}\cos\frac{(2n+1)\pi x}{2\ell}\,. \tag{70}$$

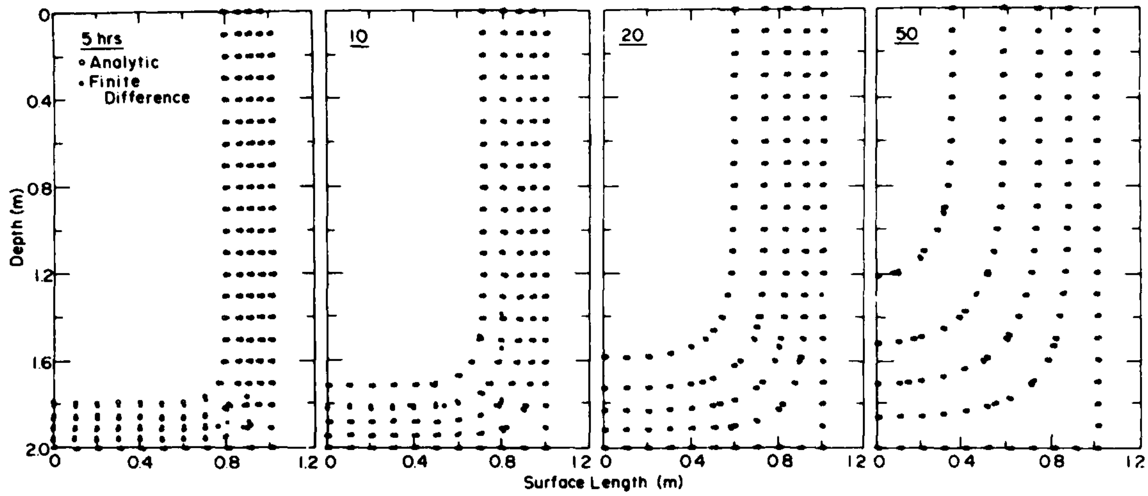
The product solution is then given by

$$T(x,y) = T_0 + \frac{16(T_i-T_0)}{\pi^2}\sum_{n=0}^{\infty}\frac{(-1)^n}{2n+1}\exp\frac{-\alpha(2n+1)^2\pi^2 t}{4\ell^2}\cos\frac{(2n+1)\pi x}{2\ell}$$

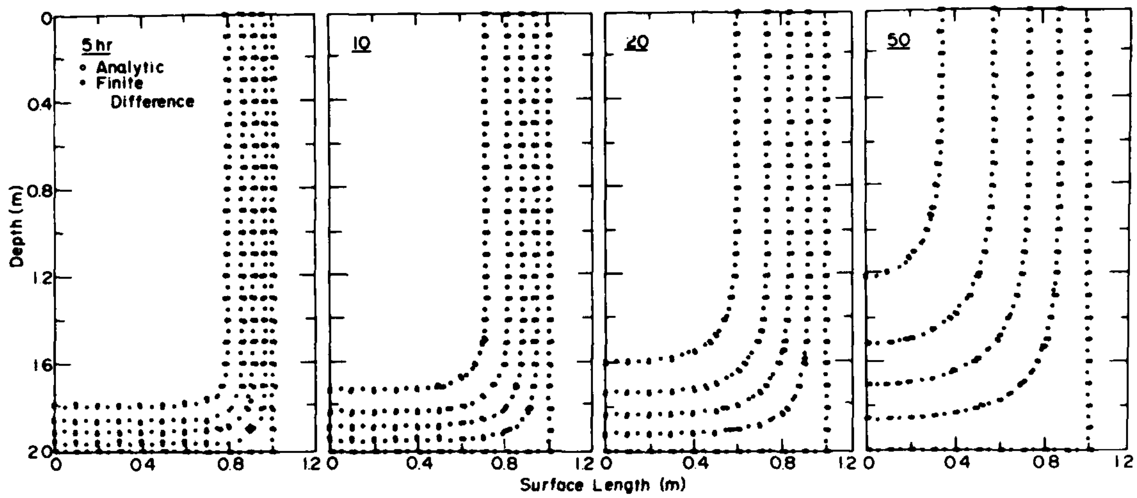
$$\cdot\sum_{m=0}^{\infty}\frac{(-1)^m}{2m+1}\exp\frac{-\alpha(2m+1)^2\pi t^2}{4b^2}\cos\frac{(2m+1)\pi x}{2b}\,. \tag{71}$$

A short FORTRAN program, INFSUM, was written to compute for this equation the various space and time increments; it is listed with the output in Appendix A. For the comparison, the rectangle is indicated by $\ell = 1$ m, $b = 2$ m, $T_0 = 40°C$, $T_i = 20°C$, and the thermal diffusivity α is set equal to 0.00251 m^2/hr.

ADI was run several times to demonstrate the effect of internodal spacing on the accuracy of the solution. Because of the symmetry of the problem, we have to model only one quarter of the problem, assigning a zero flux boundary condition to edges of the grid that fall on lines of symmetry. The region modeled is that portion of the rectangle which lies in the fourth quadrant of the Cartesian graph. The first run was made with an internodal spacing of 0.1 m and a time step of 5 hr. The resulting 24°, 28°, 32°, 36° and 40°C isotherms are plotted for several time steps (Fig. 15a). An inspection of the data printed in ADITMP reveals that the maximum discrepancy between the two solutions is 1.5°C, and it occurs in the first time step (5 hr) in the lower right-hand corner. This location at early times represents the steepest temperature gradient in the problem. By the second time step (10 hr), the discrepancy reduces to a maximum of 0.6°C. The maximum difference between the solutions continues to decrease as the temperature gradient decreases



a. Internodal spacing 0.1 m; time step 5 hr.



b. Internodal spacing 0.05 m; time step 2.5 hr.

Figure 15. Locations of 24°, 28°, 32°, 36° and 40°C isotherms (temperature increasing from left to right) from ADI (finite difference) and analytical solution.

and the location of the maximum temperature difference changes to the locations of the steepest gradients. The input and output for this run of ADI are listed in Appendix A.

The same problem was run with an internodal spacing of 0.05 m and a time step of 2.5 hours. The resulting isotherm plots are shown in Figure 15b. A closer agreement was found between the results of ADI and the analytical solution. A node-by-node comparison shows the maximum difference to be 1°C for the 5-hr distribution, decreasing to 0.4° in the 10 hr distribution. In general, the accuracy of the solution is improved as the internodal spacing decreases; the user must determine the accuracy demanded.

One-dimensional semi-infinite problem

It is also of interest to compare the results of ADI to the one-dimensional problem of a medium initially at a uniform temperature; the surface temperature then undergoes a step change to a different temperature, and the resulting temperature distribution is examined over time. This

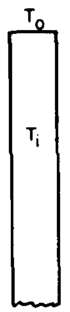


Figure 16. One-dimensional semi-infinite problem.

problem corresponds to the theoretical setup of the experimental problem examined in the *Comparison of ADI with Experimental Results* section. The problem is illustrated in Figure 16.

The solution to this analytical problem is found in many texts on heat transfer; the reader may refer to Holman (1972). The solution is given as

$$\frac{T(x,t) - T_0}{T_i - T_0} = \operatorname{erf} \frac{x}{2\sqrt{\alpha t}} \tag{72}$$

where x = the distance below the surface
t = time
α = the thermal diffusivity, 0.00251 m^2/hr in this example.

The grid used for this was 3 by 40 nodes, with an internodal spacing of 0.025 m and a time step of 0.25 hr.

For two depths, 0.05 and 0.20 m, the results of this analytical solution are compared to the results of ADI over 9 hr. The results are given in Table 1.

Table 1. Comparison of ADI results to analytical results for the one-dimensional semi-infinite problem.

x *(m)*	t *(hr)*	T *(x, t)*	*ADI*
0.05	0.5	26.42	26.66
	1.0	29.12	29.16
	2.0	31.42	31.41
	3.0	32.51	32.51
	4.0	33.18	33.18
	5.0	33.65	33.65
	6.0	34.00	34.00
	7.0	34.27	34.27
	8.0	34.50	34.49
	9.0	34.68	34.68
0.20	0.5	21.11	21.12
	1.0	21.19	21.21
	2.0	21.88	21.90
	3.0	22.83	22.85
	4.0	23.75	23.76
	5.0	24.56	24.56
	6.0	25.26	25.27
	7.0	28.88	25.88
	8.0	26.42	26.42
	9.0	26.89	26.89

The results compare to within 0.02°C for times after 2 hr. Earlier times have temperatures that differ by as much as 0.24°C near the top of the grid. Probably, this is because the steepest temperature gradients occur in this problem soon after the step change in temperature occurs at the surface. Recall that steeper temperature gradients require a smaller internodal distance; if we want greater accuracy, the model could be again run using an internodal spacing of less than 0.025 m. The smaller the internodal distance, the more accurate the solution.

Comparison of ADI with experimental results

As an example of the use of ADI, let us compare it to some experimental results. Data are available from a full-scale experiment that was conducted in a 4- by 5-m container of soil that was 1.2 m deep. Of the 4-m width, only the center 1.76 m was included in the experiment; the rest was outside an insulated boundary and was put there for support. The soil used was Lebanon sand of 19.7% moisture content. The experiment was designed to start at an initial uniform temperature of 70°F (21.11°C) in the sand in the box, then the surface temperature was to be raised 100°F (37.78°C), and the change in the temperature distribution over time was to be monitored. An analytical solution is available for this case, and ADI was also run for this case. The results compare very well (see the *One-Dimensional Semi-infinite Problem* section, analytical comparisons).

In the comparison of the experimental data to calculated results, temperatures were used that represent a 20-cm wide band taken vertically through the center of the box. Three strings of thermocouples were placed in this band, as shown in Figure 17.

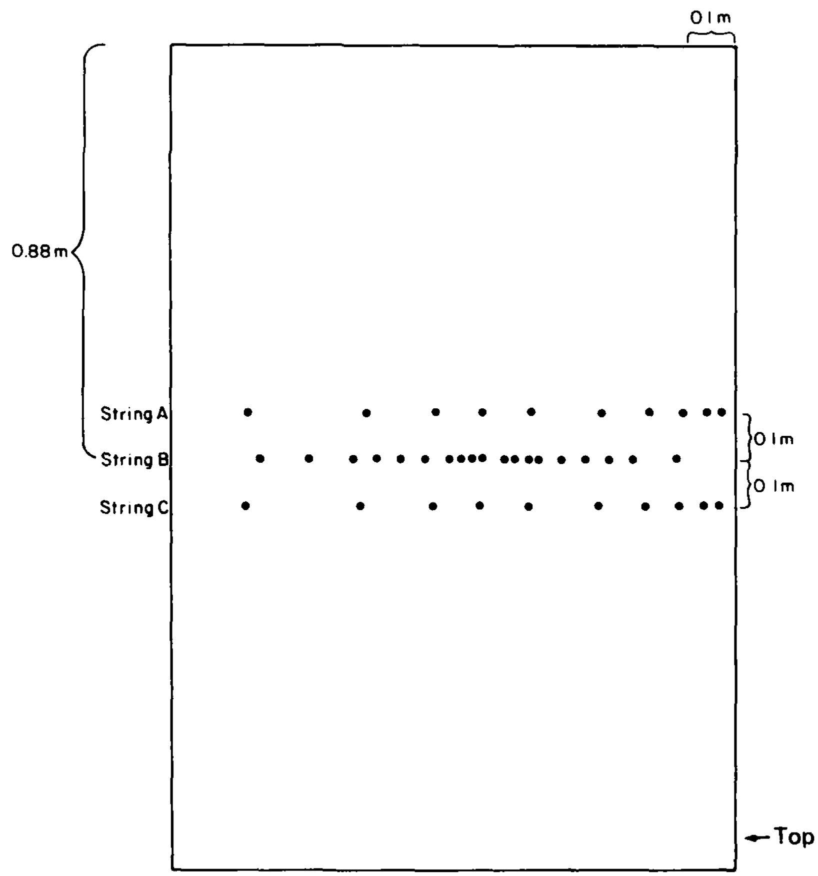


Figure 17. Thermocouple locations in the experimental setup (top of soil container on right).

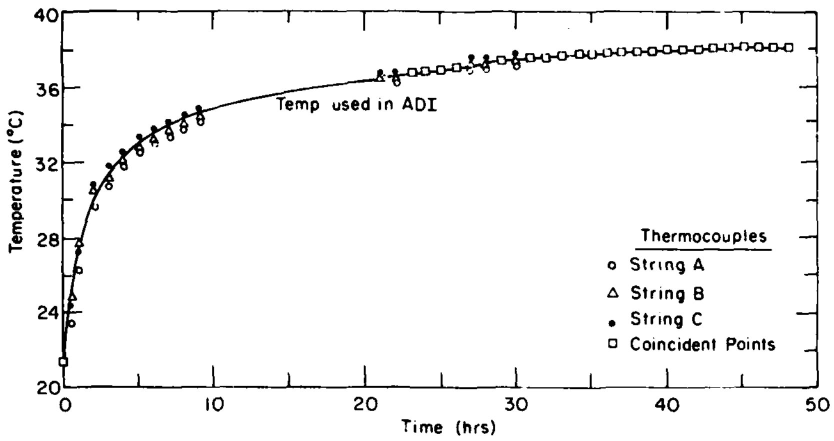


Figure 18. Surface temperatures from experiment used in ADI runs.

In the actual experiment, the initial temperatures ranged from 20.6° to 22.4°C. The initial temperature distribution for ADI was taken as an average of the three thermocouple string temperatures for each depth. It required approximately 27 hr to raise the surface temperature to 37.78°C, after which the surface temperature remained in the range from 37.2° to 38.4°C. Because the data logger malfunctioned, 12 hr of data are missing in a period beginning 9 hr after the start of the experiment; however, the rest of the equipment continued to operate normally. In comparisons made with these data it is assumed that the surface temperature increased linearly over time for this period. See Figure 18 for a graph of the surface temperature vs time for the data and that used in the comparison run with ADI.

Soil tests were conducted at CRREL to determine the density and thermal conductivity of the sand. The density was determined to be 1996.25 kg/m^3; this density was assumed constant in the ADI comparison. The conductivity was measured at two temperatures above freezing. At 4.44°C the conductivity was determined to be 1.673 W/m K, and at 26.67°C it was 1.803 W/m K. Each node in the ADI model was assigned a conductivity according to its temperature (in degrees C) for each time step from the following equation:

$$k_{i,j} = 0.00585T + 1.647 \ . \tag{73}$$

This equation was determined from the two conductivity tests.

The value of the specific heat was taken from measurements done on Lowell sand by Kersten (1949). Again, each node in the ADI model was assigned a temperature-dependent specific heat. The equation fit to Kersten's data is

$$c_{p_{i,j}} = 0.00397T + 0.34927 \ . \tag{74}$$

The 2 by 48 grid used in ADI had a distance between nodes of 0.025 m. The temperature at the surface was specified, and was taken from the data previously discussed. The time step was 0.25 hr; the surface temperature for each time step was interpolated linearly from the data. The sides and bottom of the grid were assigned the zero heat flux boundary condition. The problem was modeled for 48 hr.

The graphs in Figure 19 compare the results for several times (9 hr, 24 hr and 48 hr) during the run; they show a reasonable agreement but the model tended to underpredict the temperature change.

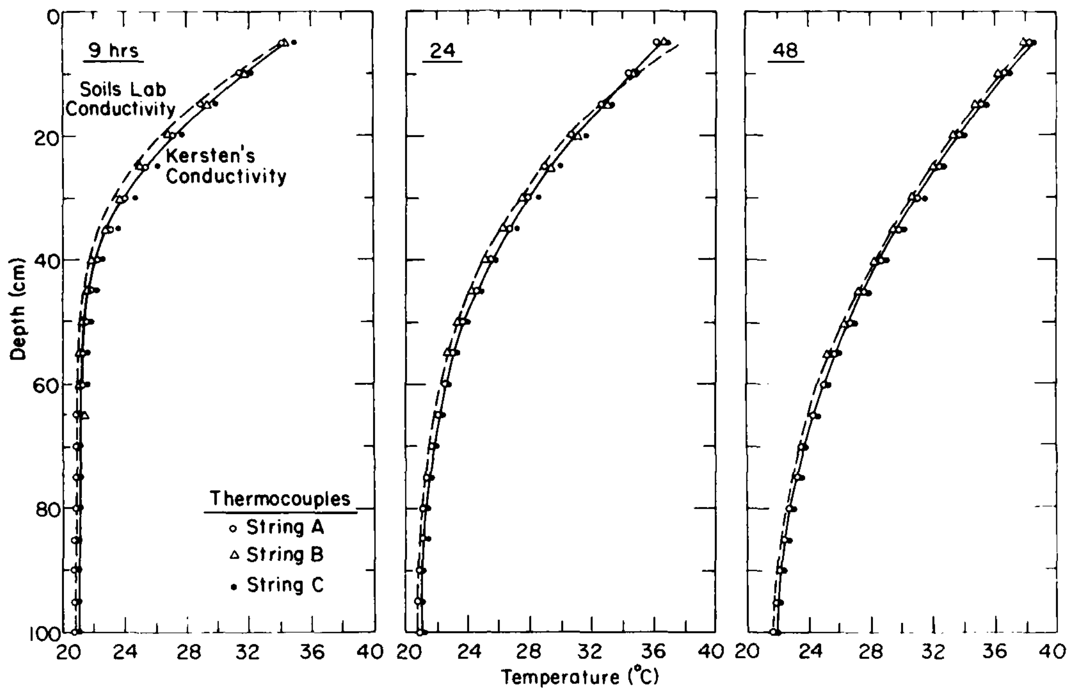


Figure 19. Results of two ADI runs using different conductivities plotted against experimental results.

The model was then run again, using conductivity measurements on Lowell sand by Kersten (1949). The temperature-dependent equation for conductivity used for this run was

$$k_{i,j} = 0.00585T + 1.975 \ . \tag{75}$$

These results may again be found in Figure 19. This time excellent agreement was found between the actual data and the ADI calculations.

Each of the two runs of ADI required 2 minutes and 13 seconds of computer time on the PRIME computer at CRREL.

VERIFICATION OF ADIPC

Comparison of ADIPC with analytical results—the Neumann solution

ADIPC was first compared to the well-known one-dimensional analytical Neumann solution. A semi-infinite region is initially at a uniform temperature T_0 which is above the fusion temperature T_f. Suddenly, the surface temperature is changed to T_s, a temperature below the fusion temperature. The subsequent movement of the phase change front may be calculated.

Allow ρ, C_p, k and H_L to represent the density, specific heat, thermal conductivity and latent heat of fusion, respectively. Thermal diffusivity α is defined as $k/\rho C_p$. Temperature T is a function of position and time. Subscripts 1 and 2 refer to the property or variable in the frozen and liquid phase, respectively. The boundary conditions follow:

$$T_1 \ (0,t) = T_s \tag{76}$$

$$T_2 \ (x,0) = T_0 \tag{77}$$

$$T_1\,(X,t) = T_2\,(X,t) = T_f \tag{78}$$

$$k_1 \frac{\partial T_1}{\partial X} - k_2 \frac{\partial T_2}{\partial X} = \rho H_L \frac{\partial X}{\partial t}\,. \tag{79}$$

The temperatures in the liquid and solid regions must satisfy the following equations

$$\frac{\partial^2 T_1}{\partial x^2} - \frac{1}{k_1}\frac{\partial T_1}{\partial x} = 0 \tag{80}$$

$$\frac{\partial^2 T_2}{\partial x^2} - \frac{1}{k_2}\frac{\partial T_2}{\partial x} = 0\,. \tag{81}$$

The solution has been given many times in the literature (e.g. Carslaw and Jaeger 1959). The solution for the location of the phase change front follows:

$$X = 2\lambda\,\sqrt{\alpha_1 t} \tag{82}$$

where λ must be determined from the relation

$$\frac{e^{-\lambda^2}}{\operatorname{erf}\lambda}\;\frac{k_2\sqrt{\alpha_1}\,(T_0 - T_f)\,e^{-\alpha_1/\alpha_2\,\lambda^2}}{k_1\sqrt{\alpha_2}\,(T_f - T_s)\operatorname{erfc}\lambda\sqrt{\alpha_1/\alpha_2}} = \frac{\lambda H_L\sqrt{\pi}}{C_{P_1}\,(T_f - T_s)}\,. \tag{83}$$

For comparison the following values were used:

$k_1 = 2.21\ \frac{\mathrm{W}}{\mathrm{m\,K}}$ $\qquad$ $k_2 = 0.580\ \frac{\mathrm{W}}{\mathrm{m\,K}}$

$\rho_1 = 917\ \frac{\mathrm{kg}}{\mathrm{m}^3}$ $\qquad$ $\rho_2 = 998.2\ \frac{\mathrm{kg}}{\mathrm{m}^3}$

$C_{P_1} = 0.5815\ \frac{\mathrm{W\,hr}}{\mathrm{kg\,K}}$ $\qquad$ $C_{P_2} = 1.16\ \frac{\mathrm{W\,hr}}{\mathrm{kg\,K}}$

$H_L = 93.00\ \frac{\mathrm{W\,hr}}{\mathrm{kg}}$

$T_f = 0°\mathrm{C}$.

$T_s = -4.67°\mathrm{C}$

$T_0 = 4.67°\mathrm{C}$

The problem was modeled for 3 hr using an internodal spacing of 0.5 cm and a time step of 0.0025 hr. The depth of the 0°C isotherm is plotted with the analytical solution in Figure 20. As mentioned earlier, the location of the phase change front is found by interpolating between the nodal temperatures to find the 0°C isotherm. The front progresses in a step-like pattern. This occurs when the location of phase change moves from one node to the next and is inherent in the apparent heat capacity method in a discretized space. Nevertheless, the results of ADIPC show good agreement with the analytical solution. A more accurate computed solution could be obtained by using a smaller internodal spacing and smaller time step. A copy of ADIPC and output for this solution is included in Appendix B.

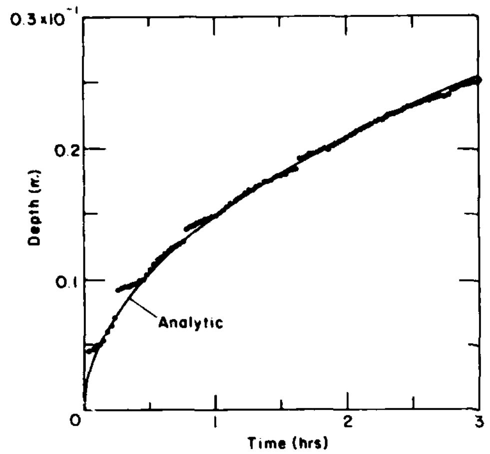


Figure 20. Comparison of ADIPC with Neumann solution.

Comparison of ADIPC with analytical results–two-dimensional phase change verification

Next compare the results of ADIPC with an analytical solution involving phase change around a pipe. This problem is two-dimensional in Cartesian coordinates and thus is a two-dimensional verification of ADIPC. But it is a one-dimensional problem in cylindrical coordinates, facilitating an analytical solution. An exact analytical solution is available for the case of freezing in a region, initially at a uniform temperature, that is suddenly subject to the effects of a continuous line source that extracts heat at a rate of Q per unit time. The exact solution of this heat conduction problem is given in Carslaw and Jaeger (1959). The results are as follows.

The location of the freezing front at any time is given by

$$R = 2\lambda\sqrt{\alpha_1 t} \tag{84}$$

where R is the radius of the front ($r = 0$ is the location of the line source), α_1 is the thermal diffusivity of the frozen zone, t is time and λ is obtained from the following relation:

$$\frac{Q}{4\pi}\exp(-\lambda^2) + \frac{k_2 (T_i - T_f)}{E_i(-\lambda\,\alpha_1/\alpha_2)}\exp(-\lambda^2\,\alpha_1/\alpha_2) = \lambda^2\,x_1\,L\rho \tag{85}$$

where k = conductivity
α = diffusivity
T_i = initial temperature
T_f = fusion temperature
$L\rho$ = latent heat per unit volume.

E_i is the exponential intergral function

$$E_i(x) = \int_{-\infty}^{x} \frac{e^v dV}{V}\,.$$

and the subscripts 1 and 2 represent the frozen and unfrozen zones, respectively.

The temperatures in the frozen and unfrozen zones are given by

$$T_1 = T_f + \frac{Q}{4\pi k_1}\left[E_i\left(-\frac{r^2}{4\alpha_1 t}\right) - E_i(-\lambda^2)\right] \qquad 0 < r < R \tag{86}$$

$$T_2 = T_i - \frac{(T_i - T_f)}{E_i(-\lambda^2 \alpha_1/\alpha_2)} \; E_i\left(\frac{-r^2}{4\alpha_2 t}\right) \qquad r > R . \tag{87}$$

In order to get a solution for phase change around a pipe from this, the pipe is assigned a constant radius r_p; the temperature at r_p varies with time according to eq 86. A shifted time is used so that at time $t = 0$ in the computer run, the location of the phase change front is at $r = r_p$. For this time, the initial temperature distribution is figured from eq 87.

For the comparison, the following values were used:

$$k_1 = 0.0072 \; \frac{\text{cal}}{\text{cm s °C}} \qquad k_2 = 0.0042 \; \frac{\text{cal}}{\text{cm s °C}}$$

$$\alpha_1 = 0.014165 \; \frac{\text{cm}^2}{\text{s}} \qquad \alpha_2 = 0.005556 \; \frac{\text{cm}^2}{\text{s}}$$

$$L\rho = 33.012 \; \frac{\text{cal}}{\text{cm}^2 \text{ °C}}$$

$$T_i = 4\text{°C}$$

$$T_f = 0\text{°C} .$$

Densities for all regions were assumed equal in the solution.

Equation 84 was solved by substituting a polynomial approximation for the exponential integrals (Abramowitz and Stegun 1970) and then solving the equation for λ by an iterative scheme on the computer. For this case it was found that $\lambda = 0.08246$.

ADIPC used an internodal distance of 1 cm and a time step of 60 s on a 30 by 30 grid. Because of the symmetry of the problem, with the Cartesian origin at the center of the pipe, only one-quarter of the situation was modeled. The top and right-hand edges of the grid were assigned zero flux boundaries, and the left-hand and bottom edges of the grid were assigned the semi-infinite condition.

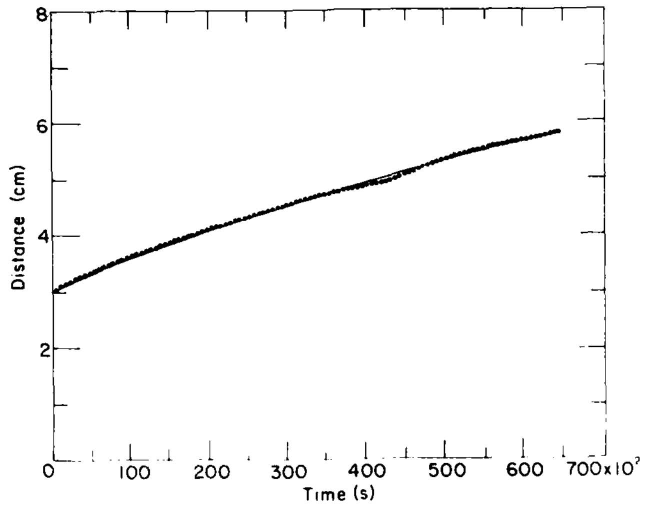


Figure 21. Comparison of ADIPC with radial analytic freezing solution.

The location of the freezing front over time was found in ADIPC by interpolating to find the zero degree isotherm; this is plotted against the analytical solution in Figure 21. Excellent agreement is found.

USER INSTRUCTION FOR ADI

The first step in using ADI to solve a problem is to define the boundaries of the problem and to identify the different materials in the problem. It is important to know the dimensions of objects to be modeled as accurately as possible. Next, identify the boundary conditions in the problem. Now draw the grid, determining a nodal spacing that will accurately represent the materials in the problem. Indicate the boundary conditions on the grid, then set these conditions in the computer program by assigning the appropriate values of RAY (I, J, 2) to each node I, J in the grid; the values are listed in the comment statements at the beginning of subroutine ADDATA.

Material properties such as thermal conductivity, density, etc., must be known in consistent units. There are no dimensional constants inherent in the program, so any system may be used. For the problems presented in this report, the following units were used: thermal conductivity (W/m K), density (kg/m^3), specific heat (W hr/kg K), temperature (°C), time (hr) and distance (m). The units of the convection coefficient would be W/m^2K. Values for the conductivity, specific heat, density and convection coefficient must be specified by the user as indicated in the main program (near the start of the 2001 and 2002 loops). These values may be programmed to change with temperature or time and could conceivably be different for each node in the grid. At the time of this report's publication, the conductivity of a semi-infinite node in the program is assumed equal to that of the adjacent regular interior node.

Now edit subroutine ADDATA following the directions in the comment statements in the subroutine to initialize the variables and arrays. Note that the variables and arrays are defined at the beginning of ADDATA.

When editing subroutine ADDATA, the user will encounter a variable named ITRT, which is defined as the number of time steps before the results are printed. Setting this value as five, for example, will set a counter in the main program so that at every fifth complete time step the main program will print the temperature distribution (RAY [I, J, 1]) into file ADITMP. Similarly, variable ITPC will set a counter to call subroutine ISOTHM to locate the user-specified isotherms in the current temperature distribution. The coordinates of these isotherms are printed into file POINTS.

When ADI is run, it will interactively ask whether or not the user wishes to run subroutine ADDATA. The first time the program is run, ADDATA must be run. The variables will be put into a formatted data file, ADIDAT. If, after the program is run, a change is made in the main program but not to ADDATA, the subroutine need not be called again; the main program will read the data from ADIDAT.

Once the program has been run, the user should examine output file ADIOUT closely to be sure that the initial conditions and boundary conditions are those intended, i.e., that no mistake was made when editing ADIDAT or changing values in the main program.

USER INSTRUCTION FOR ADIPC

The use of ADIPC is the same as ADI with the exception of the specification of several phase change variables. When editing ADDATA, the user must specify the value of the latent heat of fusion, the conductivity and density of the phase change region, and the temperature range over which phase change will occur. If the units indicated in the instructions for ADI are used, the

latent heat will have the units W hr/kg, as does the specific heat. The value of the apparent specific heat (discussed in the *Phase Change* section), known as CPPC in the program, should be calculated by the user as follows:

$$CPPC = \tfrac{1}{2}\,(C_{P_B} + C_{P_A}) + \frac{H_L}{\Delta T}$$

where C_{P_B} = specific heat of the frozen material

C_{P_A} = specific heat of the unfrozen material

H_L = latent heat of fusion for the material

ΔT = temperature range around the fusion temperature where phase change occurs from

$$T_f - \frac{\Delta T}{2} \text{ to } T_f + \frac{\Delta T}{2}\,.$$

ΔT in the verifications was 1.0°C.

As in ADI, the values for density, specific heat and conductivity must be specified as indicated in the comments near the start of the main program, ADIPC.

CONCLUSIONS

Two two-dimensional finite difference computer programs have been developed to model time-varying heat conduction. Results of test runs of the programs show excellent agreement with analytical and experimental results. The programs are easily set up to model new problems, and have the capability to solve a wide variety of heat conduction problems.

LITERATURE CITED

Abramowitz, M. and I.A. Stegun (1970) Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables. National Bureau of Standards.

Bonacina, C. and Comini, G. (1973) On the solution of the nonlinear heat conduction equations by numerical methods. *International Journal of Heat and Mass Transfer,* **16**: 581–589.

Carnahan, B., H.A. Luther and J.O. Wilkes (1969) *Applied Numerical Methods.* New York: John Wiley and Sons, Inc.

Carslaw, H.S. and J.C. Jaeger (1959) *Conduction of Heat in Solids.* Oxford: The Clarendon Press.

Croft, D.R. and D.G. Lilley (1977) *Heat Transfer Calculations Using Finite Difference Equations.* London: Applied Science Publishers, Ltd.

Gerald, C.F. (1980) *Applied Numerical Analysis.* Reading, Massachusetts: Addison-Wesley.

Holman, J.P. (1972) *Heat Transfer,* 3rd ed. New York: McGraw-Hill.

Peaceman, D.W. and H.H. Rachford (1955) The numerical solution of parabolic and elliptic differential equations. *Journal of the Society of Industrial Applied Mathematicians,* **3**: 28–41.

Phetteplace, G.L., W. Wiley and M.A. Novick (1981) Losses from the Fort Wainwright heat distribution system. CRREL Special Report 81-14.

APPENDIX A: PROGRAM INFSUM AND SAMPLE INPUT AND OUTPUT FOR PROGRAM ADI

INFSUM, a short program to do infinite sum calculation for rectangle problem

```
C   INFSUM
C   THIS CALCULATES THE INFINITE SUM PRODUCT FOR THE RECTANGLE.
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      IMPLICIT INTEGER*4(I-N)
      DIMENSION TEMP(50,50)
      CALL CONTRL(2,'INFOUT',5)
      M=50
      A=.00251
      EL=1
      B=2
      TO=40.
      TI=20.0
      PI=3.141592654
      PI2=9.869604404
      DS=.1
      DELT=1
      IX=11
      IY=21
      DO 100 IT=5,M,5
      TIM=IT*DELT
      DO 200 I=1,IY
      DO 210 J=1,IX
      Y=DS*(I-1.)
      X=DS*(J-1.)
      C1=-A*PI2*TIM/(4.*EL*EL)
      C2=PI*X/(2.*EL)
      C3=-A*PI2*TIM/(4.*B*B)
      C4=PI*Y/(2.*B)
C              FIGURE SUM1
      N=1
      SUM1=0.
      DO 220 INN=1,50
      IN=INN-1
      C1A=C1*(2.*IN+1.)*(2.*IN+1.)
      C1AA=DEXP(C1A)
      C2A=C2*(2.*IN+1.)
      FRST=((N/(2.*IN+1.))*C1AA*DCOS(C2A))
      SUM1=SUM1+FRST
      N=-1*N
      IF(C1AA.LE..0001) GO TO 221
 220  CONTINJE
 221  CONTINJE
C            FIGURE SUM2
      N=1
      SUM2=0.
      DO 230 INN=1,50
      IN=INN-1
      C3A=C3*(2.*IN+1.)*(2.*IN+1.)
      C4A=C4*(2.*IN+1.)
      C3AA=DEXP(C3A)
      SCND=(N/(2.*IN+1.))*C3AA*DCOS(C4A)
      SUM2=SUM2+SCND
      N=-1*N
      IF(C3AA.LE..0001) GO TO 231
 230  CONTINJE
 231  CONTINJE
      TEMP(I,J)=TO+(TI-TO)*(16./PI2)*SUM1*SUM2
 210  CONTINUE
 200  CONTINUE
      WRITE(5,211) TIM
 211  FORMAT(/,1X,'TEMPERATURES AFTER',F6.2,' HOURS')
      WRITE(5,212) ((TEMP(I,J),J=1,IX),I=1,IY)
 212  FORMAT(1X,11F6.2)
 100  CONTINUE
      CALL CONTRL(4,'INFOUT',5)
      CALL EXIT
      END
```

Sample output from program INFSUM

```
TEMPERATURES AFTER   5.00 HOURS
 20.00  20.00  20.00  20.00  20.00  20.03  20.23  21.17  24.14  30.56  40.00
 20.00  20.00  20.00  20.00  20.00  20.03  20.23  21.17  24.14  30.56  40.00
 20.00  20.00  20.00  20.00  20.00  20.03  20.23  21.17  24.14  30.56  40.00
 20.00  20.00  20.00  20.00  20.00  20.03  20.23  21.17  24.14  30.56  40.00
 20.00  20.00  20.00  20.00  20.00  20.03  20.23  21.17  24.14  30.56  40.00
 20.00  20.00  20.00  20.00  20.00  20.03  20.23  21.17  24.14  30.56  40.00
 20.00  20.00  20.00  20.00  20.00  20.03  20.23  21.17  24.14  30.56  40.00
 20.00  20.00  20.00  20.00  20.00  20.03  20.23  21.17  24.14  30.56  40.00
 20.00  20.00  20.00  20.00  20.00  20.03  20.23  21.17  24.14  30.56  40.00
 20.00  20.00  20.00  20.00  20.00  20.03  20.23  21.17  24.14  30.56  40.00
 20.00  20.00  20.00  20.00  20.00  20.03  20.23  21.17  24.14  30.56  40.00
 20.00  20.00  20.00  20.00  20.00  20.03  20.23  21.17  24.14  30.56  40.00
 20.00  20.00  20.00  20.00  20.00  20.03  20.23  21.17  24.14  30.56  40.00
 20.00  20.00  20.00  20.00  20.00  20.03  20.23  21.17  24.14  30.56  40.00
 20.00  20.00  20.00  20.00  20.01  20.04  20.23  21.17  24.14  30.56  40.00
 20.03  20.03  20.03  20.03  20.04  20.06  20.26  21.20  24.16  30.57  40.00
 20.23  20.23  20.23  20.23  20.23  20.26  20.46  21.38  24.32  30.67  40.00
 21.17  21.17  21.17  21.17  21.17  21.20  21.38  22.26  25.06  31.11  40.00
 24.14  24.14  24.14  24.14  24.14  24.16  24.32  25.06  27.42  32.51  40.00
 30.56  30.56  30.56  30.56  30.56  30.57  30.67  31.11  32.51  35.54  40.00
 40.00  40.00  40.00  40.00  40.00  40.00  40.00  40.00  40.00  40.00  40.00

TEMPERATURES AFTER  10.00 HOURS
 20.00  20.00  20.01  20.04  20.15  20.51  21.48  23.61  27.44  33.11  40.00
 20.00  20.00  20.01  20.04  20.15  20.51  21.48  23.61  27.44  33.11  40.00
 20.00  20.00  20.01  20.04  20.15  20.51  21.48  23.61  27.44  33.11  40.00
 20.00  20.00  20.01  20.04  20.15  20.51  21.48  23.61  27.44  33.11  40.00
 20.00  20.00  20.01  20.04  20.15  20.51  21.48  23.61  27.44  33.11  40.00
 20.00  20.00  20.01  20.04  20.15  20.51  21.48  23.61  27.44  33.11  40.00
 20.00  20.00  20.01  20.04  20.15  20.51  21.48  23.61  27.44  33.11  40.00
 20.00  20.00  20.01  20.04  20.15  20.51  21.48  23.61  27.44  33.11  40.00
 20.00  20.00  20.01  20.04  20.15  20.51  21.48  23.61  27.44  33.11  40.00
 20.00  20.00  20.01  20.04  20.15  20.51  21.48  23.61  27.44  33.11  40.00
 20.00  20.00  20.01  20.04  20.15  20.51  21.48  23.61  27.44  33.11  40.00
 20.00  20.00  20.01  20.04  20.15  20.51  21.49  23.61  27.44  33.11  40.00
 20.01  20.01  20.01  20.04  20.16  20.52  21.49  23.62  27.45  33.11  40.00
 20.04  20.04  20.04  20.07  20.18  20.55  21.52  23.64  27.46  33.12  40.00
 20.15  20.15  20.15  20.18  20.30  20.66  21.62  23.73  27.53  33.16  40.00
 20.51  20.51  20.52  20.55  20.66  21.01  21.96  24.03  27.76  33.28  40.00
 21.48  21.49  21.49  21.52  21.62  21.96  22.86  24.83  28.37  33.62  40.00
 23.61  23.61  23.62  23.64  23.73  24.03  24.83  26.57  29.71  34.35  40.00
 27.44  27.44  27.45  27.46  27.53  27.76  28.37  29.71  32.11  35.67  40.00
 33.11  33.11  33.11  33.12  33.16  33.28  33.62  34.35  35.67  37.62  40.00
 40.00  40.00  40.00  40.00  40.00  40.00  40.00  40.00  40.00  40.00  40.00

TEMPERATURES AFTER  15.00 HOURS
 20.01  20.02  20.07  20.21  20.58  21.37  22.90  25.49  29.32  34.31  40.00
 20.01  20.02  20.07  20.21  20.58  21.37  22.90  25.49  29.32  34.31  40.00
 20.01  20.02  20.07  20.21  20.58  21.37  22.90  25.49  29.32  34.31  40.00
 20.01  20.02  20.07  20.21  20.58  21.37  22.90  25.49  29.32  34.31  40.00
 20.01  20.02  20.07  20.21  20.58  21.37  22.90  25.49  29.32  34.31  40.00
 20.01  20.02  20.07  20.21  20.58  21.37  22.90  25.49  29.32  34.31  40.00
 20.01  20.02  20.07  20.21  20.58  21.37  22.90  25.49  29.32  34.31  40.00
 20.01  20.02  20.07  20.21  20.58  21.37  22.90  25.49  29.32  34.31  40.00
 20.01  20.02  20.07  20.22  20.58  21.37  22.90  25.49  29.32  34.31  40.00
 20.01  20.02  20.07  20.22  20.58  21.37  22.90  25.49  29.32  34.31  40.00
 20.02  20.03  20.08  20.22  20.58  21.37  22.90  25.49  29.32  34.31  40.00
 20.03  20.04  20.09  20.24  20.60  21.39  22.92  25.50  29.33  34.32  40.00
 20.08  20.09  20.14  20.29  20.64  21.43  22.96  25.54  29.36  34.33  40.00
 20.23  20.24  20.29  20.43  20.78  21.57  23.08  25.64  29.44  34.37  40.00
 20.59  20.60  20.64  20.78  21.13  21.90  23.39  25.90  29.63  34.47  40.00
 21.38  21.39  21.44  21.57  21.90  22.64  24.07  26.48  30.05  34.70  40.00
 22.91  22.92  22.96  23.08  23.39  24.07  25.38  27.59  30.87  35.14  40.00
 25.49  25.50  25.54  25.64  25.90  26.48  27.59  29.47  32.25  35.87  40.00
 29.33  29.33  29.36  29.44  29.63  30.05  30.87  32.25  34.30  36.96  40.00
 34.31  34.32  34.33  34.37  34.47  34.70  35.14  35.87  36.96  38.38  40.00
 40.00  40.00  40.00  40.00  40.00  40.00  40.00  40.00  40.00  40.00  40.00

TEMPERATURES AFTER  20.00 HOURS
 20.06  20.10  20.23  20.54  21.17  22.29  24.14  26.87  30.56  35.05  40.00
 20.06  20.10  20.23  20.54  21.17  22.29  24.14  26.87  30.56  35.05  40.00
 20.06  20.10  20.23  20.54  21.17  22.29  24.14  26.87  30.56  35.05  40.00
 20.06  20.10  20.23  20.54  21.17  22.29  24.14  26.87  30.56  35.05  40.00
 20.06  20.10  20.23  20.54  21.17  22.29  24.14  26.87  30.56  35.05  40.00
 20.06  20.10  20.23  20.54  21.17  22.29  24.14  26.87  30.56  35.05  40.00
 20.06  20.10  20.23  20.54  21.17  22.29  24.14  26.88  30.56  35.05  40.00
 20.06  20.10  20.24  20.54  21.17  22.29  24.14  26.88  30.56  35.05  40.00
 20.07  20.10  20.24  20.55  21.17  22.29  24.14  26.88  30.56  35.05  40.00
 20.07  20.11  20.24  20.55  21.18  22.30  24.14  26.88  30.56  35.05  40.00
 20.10  20.13  20.27  20.58  21.20  22.32  24.16  26.90  30.57  35.05  40.00
 20.15  20.19  20.32  20.63  21.25  22.37  24.21  26.93  30.60  35.07  40.00
 20.29  20.33  20.46  20.77  21.38  22.50  24.32  27.03  30.67  35.10  40.00
 20.61  20.64  20.77  21.07  21.68  22.77  24.57  27.23  30.81  35.18  40.00
 21.23  21.26  21.39  21.68  22.26  23.32  25.06  27.64  31.11  35.33  40.00
 22.35  22.38  22.50  22.77  23.32  24.32  25.95  28.38  31.64  35.61  40.00
 24.19  24.22  24.32  24.57  25.06  25.95  27.42  29.59  32.51  36.07  40.00
 26.92  26.94  27.03  27.23  27.64  28.38  29.59  31.39  33.80  36.75  40.00
 30.59  30.61  30.67  30.82  31.11  31.64  32.51  33.80  35.54  37.66  40.00
 35.06  35.07  35.10  35.18  35.33  35.61  36.07  36.75  37.66  38.77  40.00
 40.00  40.00  40.00  40.00  40.00  40.00  40.00  40.00  40.00  40.00  40.00
```

Data printed by program ADI for rectangle problem (file ADIOUT)

```
DATA FOR THIS RUN OF ADI2D:

    DS         DELT         DI
  0.10000    5.00000   50.00000

   A     X      Y NISO ITRT IMAX
   1    11     21    5    1   10

TISO(K),K=1,NISO:
  40.00
  36.00
  32.00
  28.00
  24.00

KRNR(1)= 2 KRNR(2)= 1  KRNR(3)= 1  KRNR(4)= 1

        FLXT(J)          FLXB(J)          TMPT(J)          TMPB(J)
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000

        FLXL(I)          FLXR(I)          TMPL(I)          TMPR(I)
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000
         0.0000           0.0000          20.0000          40.0000

RAY(I,J,5)    NODAL MATERIAL TYPE
    1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
    1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
    1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
    1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
    1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
    1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
    1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
    1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
    1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
    1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
    1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
    1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
    1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
    1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
    1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
    1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
    1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
    1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
    1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
    1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
    1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00

RAY(I,J,2)    NODAL LOCATION TYPE
    1.00      8.00      8.00      8.00      8.00      8.00      8.00      8.00      8.00      8.00      4.00
    8.00      5.00      5.00      5.00      5.00      5.00      5.00      5.00      5.00      5.00      7.00
    8.00      5.00      5.00      5.00      5.00      5.00      5.00      5.00      5.00      5.00      7.00
    8.00      5.00      5.00      5.00      5.00      5.00      5.00      5.00      5.00      5.00      7.00
    8.00      5.00      5.00      5.00      5.00      5.00      5.00      5.00      5.00      5.00      7.00
    8.00      5.00      5.00      5.00      5.00      5.00      5.00      5.00      5.00      5.00      7.00
    8.00      5.00      5.00      5.00      5.00      5.00      5.00      5.00      5.00      5.00      7.00
    8.00      5.00      5.00      5.00      5.00      5.00      5.00      5.00      5.00      5.00      7.00
    8.00      5.00      5.00      5.00      5.00      5.00      5.00      5.00      5.00      5.00      7.00
    8.00      5.00      5.00      5.00      5.00      5.00      5.00      5.00      5.00      5.00      7.00
```

```
    8.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     7.00
    8.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     7.00
    8.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     7.00
    8.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     7.00
    8.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     7.00
    8.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     7.00
    8.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     7.00
    8.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     7.00
    8.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     7.00
    8.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     5.00     7.00
    2.00     7.00     7.00     7.00     7.00     7.00     7.00     7.00     7.00     7.00     3.00

RAY(I,J,1)    TEMPERATURES AT THE START:
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   40.00    40.00    40.00    40.00    40.00    40.00    40.00    40.00    40.00    40.00    40.00
TEMPERATURES WILL BE PRINTED EVERY    1 TIME STEPS.

 BOUNDARY CONDITIONS:

TOP BOUNDARY          CONSTANT HEAT FLUX
TOP LEFT CORNER       CONST FLUX ON BOTH SIDES
LEFT BOUNDARY         CONSTANT HEAT FLUX
BOTTOM LEFT CORNER CONSTANT TEMPERATURE
BOTTOM BOUNDARY       CONSTANT TEMPERATURE
BOTTOM RIGHT CORNERCONSTANT TEMPERATURE
RIGHT BOUNDARY        CONSTANT TEMPERATURE
TOP RIGHT CORNER      CONSTANT TEMPERATURE

NUMBER OF TIME STEPS=    10
```

Sample output from program ADI for rectangle problem (file ADITMP)

```
RAY(I,J,1)    TEMPERATURES AT THE START:
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    20.00    40.00
   40.00    40.00    40.00    40.00    40.00    40.00    40.00    40.00    40.00    40.00    40.00

TEMPERATURES AFTER    1 TIME STEPS (    5.00 HRS):
   20.00    20.00    20.00    20.01    20.03    20.10    20.34    21.12    23.70    32.16    40.00
   20.00    20.00    20.00    20.01    20.03    20.10    20.34    21.12    23.70    32.16    40.00
   20.00    20.00    20.00    20.01    20.03    20.10    20.34    21.12    23.70    32.16    40.00
   20.00    20.00    20.00    20.01    20.03    20.10    20.34    21.12    23.70    32.16    40.00
   20.00    20.00    20.00    20.01    20.03    20.10    20.34    21.12    23.70    32.16    40.00
   20.00    20.00    20.00    20.01    20.03    20.10    20.34    21.12    23.70    32.16    40.00
   20.00    20.00    20.00    20.01    20.03    20.10    20.34    21.12    23.70    32.16    40.00
   20.00    20.00    20.00    20.01    20.03    20.10    20.34    21.12    23.70    32.16    40.00
   20.00    20.00    20.00    20.01    20.03    20.10    20.34    21.12    23.70    32.16    40.00
   20.00    20.00    20.00    20.01    20.03    20.10    20.34    21.12    23.70    32.16    40.00
   20.00    20.00    20.00    20.01    20.03    20.10    20.34    21.12    23.70    32.16    40.00
   20.00    20.00    20.00    20.01    20.03    20.10    20.34    21.13    23.70    32.16    40.00
   20.00    20.00    20.01    20.01    20.03    20.11    20.34    21.13    23.70    32.16    40.00
```

20.01	20.01	20.01	20.02	20.04	20.11	20.35	21.13	23.71	32.17	40.00
20.03	20.03	20.03	20.04	20.06	20.14	20.37	21.15	23.72	32.17	40.00
20.10	20.10	20.11	20.11	20.14	20.21	20.44	21.22	23.78	32.20	40.00
20.34	20.34	20.34	20.35	20.37	20.44	20.66	21.45	23.98	32.30	40.00
21.12	21.13	21.13	21.13	21.15	21.22	21.45	22.19	24.61	32.60	40.00
23.70	23.70	23.70	23.71	23.72	23.78	23.98	24.61	26.71	33.61	40.00
32.16	32.16	32.16	32.17	32.17	32.20	32.30	32.60	33.61	36.93	40.00
40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00

TEMPERATURES AFTER 2 TIME STEPS (10.00 HRS):

20.01	20.01	20.03	20.07	20.20	20.55	21.46	23.60	27.89	32.98	40.00
20.01	20.01	20.03	20.07	20.20	20.55	21.46	23.60	27.89	32.98	40.00
20.01	20.01	20.03	20.07	20.20	20.55	21.46	23.60	27.89	32.98	40.00
20.01	20.01	20.03	20.07	20.20	20.55	21.45	23.60	27.89	32.98	40.00
20.01	20.01	20.03	20.07	20.20	20.55	21.46	23.60	27.89	32.98	40.00
20.01	20.01	20.03	20.07	20.20	20.55	21.46	23.60	27.89	32.98	40.00
20.01	20.01	20.03	20.07	20.20	20.55	21.46	23.60	27.89	32.98	40.00
20.01	20.01	20.03	20.07	20.20	20.55	21.45	23.60	27.89	32.98	40.00
20.01	20.01	20.03	20.07	20.20	20.56	21.46	23.60	27.89	32.98	40.00
20.01	20.01	20.03	20.07	20.20	20.56	21.46	23.60	27.89	32.98	40.00
20.01	20.01	20.03	20.07	20.21	20.56	21.46	23.60	27.90	32.98	40.00
20.01	20.02	20.03	20.08	20.21	20.56	21.47	23.61	27.90	32.98	40.00
20.03	20.03	20.05	20.13	20.23	20.58	21.48	23.62	27.91	32.99	40.00
20.08	20.08	20.10	20.14	20.27	20.62	21.53	23.66	27.94	33.01	40.00
20.21	20.21	20.23	20.27	20.40	20.75	21.65	23.77	28.02	33.05	40.00
20.56	20.56	20.58	20.62	20.75	21.09	21.97	24.05	28.23	33.18	40.00
21.46	21.47	21.48	21.53	21.65	21.97	22.81	24.80	28.78	33.49	40.00
23.60	23.61	23.62	23.66	23.77	24.05	24.80	26.55	30.07	34.24	40.00
27.90	27.90	27.91	27.94	28.02	28.23	28.78	30.07	32.67	35.75	40.00
32.98	32.98	32.99	33.01	33.05	33.18	33.49	34.24	35.75	37.54	40.00
40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00

TEMPERATURES AFTER 3 TIME STEPS (15.00 HRS):

20.03	20.04	20.10	20.25	20.61	21.39	22.94	25.60	29.30	34.45	40.00
20.03	20.04	20.10	20.25	20.61	21.39	22.94	25.60	29.30	34.45	40.00
20.03	20.04	20.10	20.25	20.61	21.39	22.94	25.60	29.30	34.45	40.00
20.03	20.04	20.10	20.25	20.61	21.39	22.94	25.60	29.30	34.45	40.00
20.03	20.04	20.10	20.25	20.61	21.39	22.94	25.60	29.30	34.45	40.00
20.03	20.04	20.10	20.25	20.61	21.39	22.94	25.60	29.30	34.45	40.00
20.03	20.04	20.10	20.25	20.61	21.39	22.94	25.60	29.30	34.45	40.00
20.03	20.04	20.10	20.25	20.61	21.39	22.94	25.60	29.30	34.45	40.00
20.03	20.05	20.10	20.25	20.61	21.39	22.94	25.60	29.30	34.45	40.00
20.03	20.05	20.11	20.26	20.61	21.39	22.94	25.60	29.30	34.45	40.00
20.04	20.06	20.12	20.27	20.62	21.40	22.95	25.61	29.31	34.45	40.00
20.07	20.08	20.14	20.29	20.65	21.42	22.97	25.63	29.32	34.46	40.00
20.13	20.14	20.20	20.35	20.70	21.48	23.02	25.67	29.35	34.48	40.00
20.28	20.30	20.35	20.50	20.85	21.62	23.15	25.78	29.44	34.52	40.00
20.64	20.65	20.71	20.85	21.20	21.95	23.45	26.04	29.63	34.62	40.00
21.41	21.43	21.45	21.62	21.95	22.68	24.12	26.60	30.04	34.83	40.00
22.96	22.97	23.02	23.15	23.45	24.12	25.44	27.71	30.87	35.26	40.00
25.62	25.63	25.67	25.78	26.04	26.60	27.71	29.63	32.30	36.00	40.00
29.32	29.32	29.34	29.44	29.63	30.04	30.97	32.30	34.28	37.03	40.00
34.46	34.46	34.48	34.52	34.62	34.83	35.26	36.00	37.03	38.46	40.00
40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00	40.00

Sample of isotherm locations output from ADI for rectangle problem (file POINTS)

```
  5.00
THE FOLLOWING  144 POINTS REPRESENT TIME=    5.00 HOURS:
          40.00 DEGREE ISOTHERM:
           0.000           2.000
           0.100           2.000
           0.200           2.000
           0.300           2.000
           0.400           2.000
           0.500           2.000
           0.600           2.000
           0.700           2.000
           0.800           2.000
           0.900           2.000
           1.000           0.100
           1.000           0.200
           1.000           0.300
           1.000           0.400
           1.000           0.500
           1.000           0.600
           1.000           0.700
           1.000           0.800
           1.000           0.900
           1.000           1.000
           1.000           1.100
           1.000           1.200
           1.000           1.300
           1.000           1.400
           1.000           1.500
           1.000           1.600
           1.000           1.700
           1.000           1.800
           1.000           1.900
           1.000           2.000
```

36.00 DEGREE ISOTHERM:

0.949	0.100
0.949	0.200
0.949	0.300
0.949	0.400
0.949	0.500
0.949	0.600
0.949	0.700
0.949	0.800
0.949	0.900
0.949	1.000
0.949	1.100
0.949	1.200
0.949	1.300
0.947	1.400
0.949	1.500
0.948	1.600
0.946	1.700
0.937	1.800
0.872	1.900
0.900	1.872
0.000	1.949
0.100	1.949
0.200	1.949
0.300	1.949
0.400	1.949
0.500	1.949
0.600	1.946
0.700	1.946
0.800	1.937
0.949	0.000

32.00 DEGREE ISOTHERM:

0.898	0.100
0.898	0.200
0.898	0.300
0.898	0.400
0.898	0.500
0.898	0.600
0.898	0.700
0.898	0.800
0.898	0.900
0.898	1.000
0.898	1.100
0.898	1.200
0.898	1.300
0.896	1.400
0.895	1.500
0.896	1.600
0.892	1.700
0.877	1.800
0.000	1.898
0.100	1.898
0.200	1.898
0.300	1.898
0.400	1.898
0.500	1.898
0.600	1.896
0.700	1.892
0.800	1.877
0.898	0.000

APPENDIX B: PROGRAM ADIPC AND SAMPLE INPUT AND OUTPUT

Program ADIPC and its subroutines

```
 1)
 2) C ************************************************************************
 3) C ************************************************************************
 4) C                               ADIPC
 5) C
 6) C    THIS FORTRAN PROGRAM SOLVES FOR TWO DIMENSIONAL TEMPERATURE
 7) C   DISTRIBUTION RESULTING FROM CONDUCTION HEAT TRANSFER. BOUNDARY CONDITIONS
 8) C   MAY INCLUDE SPECIFIED TEMPERATURES, CONVECTIVE SURFACES, AND
 9) C   SEMI-INFINITE BOUNDARIES. ADIPC USES AN IMPLICIT ALTERNATING DIRECTION
10) C   FINITE DIFFERENCE NUMERICAL TECHNIQUE ON A GRID WITH SQUARE ELEMENTS.
11) C   THE MATERIAL PROPERTIES (TK,CP,RO,H) MAY BE DIFFERENT FOR EACH
12) C   NODE OF THE GRID, AND MAY CHANGE WITH TIME.
13) C   ADIPC USES THE APPARENT HEAT CAPACITY METHOD FOR PHASE CHANGE.
14) C    THIS PROGRAM WAS WRITTEN BY MARY REMLEY ALBERT AT CRREL, 1981.
15)
16) C   DATA FOR ADIPC IS GATHERED BY ADDATA, WHICH PUTS IT INTO FILE ADPDAT.
17) C   SEE ADDATA FOR AN EXPLANATION OF VARIABLES, AND TO SET UP THE PROGRAM
18) C   FOR YOUR PROBLEM.
19) C   ADJUSTMENTS FOR CHANGING MATERIAL PROPERTIES WITH TIME (DURING THE RUN)
20) C   SHOULD BY TYPED INTO THE MAIN PROGRAM AS INDICATED NEAR LINE 170.
21) C
22) C ********************************************************************
23) C ********************************************************************
24) C
25) C
26) C              DO NOT CHANGE THE NEXT 130 LINES.
27)       IMPLICIT INTEGER*2(A,B,I-L,N,Q,W,X,Y,Z)
28)       IMPLICIT DOUBLE PRECISION(C-H,M,O,P,R-V)
29) C    M1 IS THE COMMON LOCATION FOR FOR MAIN AND ADDATA VARIABLES
30) C    M12 IS THE COMMON LOCATION FOR FOR MAIN, ADDATA,AND ISOTHM VARIABLES
31)       COMMON/M12I/ X,Y,NISO,Q
32)       COMMON/M12R/ RAY(80,80,3),TISO(9),DS,DELT
33)       COMMON/M1I/ IMAX,A,ITRT,KRNR(4),ITPC
34)       COMMON/M1R/ DI,TDEL,H(2),FLXT(80),
35)      & FLXB(80),FLXL(80),FLXR(80),TMPT(80),
36)      & TMPB(80),TMPL(80),TMPR(80),ROPC(2),CPPC(2),HL(2),TPC(2)
37)       COMMON/MMM/ TEMP(80,80),EA(200),
38)      & EB(200),C(200),D(200),CP(80,80),TK(80,80),
39)      & RO(80,80),OLDT(80,80),CPO(80,80),ROO(80,80),
40)      & IFLAG(10),ISTAT(80,80)
41) C            OPEN FILES
42)       CALL CONTRL(3,'ADPDAT',5)
43)       CALL CONTRL(2,'ADPOUT',6)
44) C     FUNITS 7 & 8 USED IN ISOTHM. FUNIT 10 NOT USABLE
45)       CALL CONTRL(2,'ADPNDT',9)
46)       CALL CONTRL(2,'ADPTMP',11)
47)       CALL CONTRL(2,'POINT1',13)
48)       WRITE(1,1)
49)  1    FORMAT(1X,'IF YOU CHANGED ADDATA SINCE YOU LAST RAN THIS,',
50)      C ' TYPE "1"',/,1X,'AND ADDATA WILL BE EXECUTED.',/,1X,
51)      C 'OTHERWISE, TYPE "2"')
52)       READ(1,*) IT
53)       IF(IT.EQ.2) GO TO 4
54)       CALL ADDATA
55)       GO TO 12
56)  4    CONTINUE
57) C            READ DATA INPUT FROM FILE ADPDAT
58)       READ(5,10) DS,DELT,DI,TDEL
59)  10   FORMAT(1X,4F10.5)
60)       READ(5,20) A,X,Y,NISO,ITRT,IMAX,ITPC
61)  20   FORMAT(1X,7I5)
62)       READ(5,16) ((ROPC(K),CPPC(K),HL(K),TPC(K)),K=1,A)
63)  16   FORMAT(1X,4F10.5)
64)       READ(5,15)(TISO(B),B=1,NISO)
65)  15   FORMAT(1X,F7.2)
66)       READ(5,30) (H(L),L=1,A)
67)  30   FORMAT(1X,F10.5)
68)       READ(5,35) ((RAY(I,J,3),J=1,X),I=1,Y)
69)       READ(5,35) ((RAY(I,J,2),J=1,X),I=1,Y)
70)       READ(5,35) ((RAY(I,J,1),J=1,X),I=1,Y)
71)  35   FORMAT(1X,17F7.2)
72)       READ(5,34) (KRNR(J),J=1,4)
73)  34   FORMAT(1X,4I2)
74)       READ(5,35) (FLXT(J),J=1,X)
75)       READ(5,35) (FLXB(J),J=1,X)
76)       READ(5,35) (FLXL(I),I=1,Y)
77)       READ(5,35) (FLXR(I),I=1,Y)
78)       READ(5,35) (TMPT(J),J=1,X)
79)       READ(5,35) (TMPB(J),J=1,X)
80)       READ(5,35) (TMPL(I),I=1,Y)
```

```
      READ(5,35) (TMPR(I),I=1,Y)
   12 CONTINUE
C        WRITE INITIAL DATA INTO OUTPUT FILE ADPOUT
      WRITE(6,21)
   21 FORMAT(/,1X,'DATA FOR THIS RUN OF ADIPC:',/,/)
      WRITE(6,22)
   22 FORMAT(/,1X,'     DS       DELT        DI        TDEL')
      WRITE(6,10) (DS,DELT,DI,TDEL)
      WRITE(6,23)
   23 FORMAT(/,5X,'A',4X,'X',4X,'Y',1X,'NISO ITRT IMAX ITPC ')
      WRITE(6,20)(A,X,Y,NISO,ITRT,IMAX,ITPC)
      WRITE(6,29)
   29 FORMAT(/,1X,'  ROPC(K)    CPPC(K)   HL(K)     TPC(K)')
      WRITE(6,16) ((ROPC(K),CPPC(K),HL(K),TPC(K)),K=1,A)
      WRITE(6,24)
   24 FORMAT(/,1X,'TISO(B),B=1,NISO:')
      WRITE(6,15)(TISO(B),B=1,NISO)
      WRITE(6,25)
   25 FORMAT(/,3X,'H(K)')
      WRITE(6,30) (H(L),L=1,A)
      WRITE(6,33) (KRNR(J),J=1,4)
   33 FORMAT(/,1X,'KRNR(1)=',I2,' KRNR(2)=',I2,'  KRNR(3)=',I2,
     & '  KRNR(4)=',I2)
      WRITE(6,26)
   26 FORMAT(/,3X,'FLXT(J)',3X,'FLXB(J)',3X,'TMPT(J)',3X,
     & 'TMPB(J)')
      WRITE(6,27)((FLXT(J),FLXB(J),TMPT(J),TMPB(J)),J=1,X)
   27 FORMAT(1X,4F10.2)
      WRITE(6,28)
   28 FORMAT(/,3X,'FLXL(I)',3X,'FLXR(I)',3X,'TMPL(I)',3X,
     & 'TMPR(I)')
      WRITE(6,27) ((FLXL(I),FLXR(I),TMPL(I),TMPR(I)),I=1,Y)
      RRX=X/17
      JJX=RRX
      JJR=X-(17*JJX)
      DO 63 K=1,3
      GO TO (64,65,66),K
   64 WRITE(6,31)
      GO TO 67
   65 WRITE(6,45)
      GO TO 67
   66 WRITE(6,44)
   67 DO 68 JJ=1,JJX
      IF(JJX.EQ.0) GO TO 68
      J2=17*JJ
      J1=J2-16
      WRITE(6,500) ((RAY(I,J,K),J=J1,J2),I=1,Y)
      IF(JJ.LT.JJX) WRITE(6,503)
   68 CONTINUE
      J2=X
      J1=X-JJR+1
      IF(JJR.EQ.0) GO TO 69
      IF(JJX.NE.0) WRITE(6,503)
      DO 69 I=1,Y
      WRITE(6,500) (RAY(I,J,K),J=J1,J2)
   69 CONTINUE
   63 CONTINUE
   44 FORMAT(/,1X,'RAY(I,J,3)   NODAL MATERIAL TYPE')
   45 FORMAT(/,1X,'RAY(I,J,2)   NODAL LOCATION TYPE')
   31 FORMAT(/,1X,'RAY(I,J,1)   TEMPERATURES AT THE START:')
      WRITE(1,11) ITRT,ITPC
      WRITE(6,11) ITRT,ITPC
   11 FORMAT(/,/,1X,'TEMPERATURES WILL BE PRINTED EVERY',I4,
     & ' TIME STEPS.',/,1X,'ISOTHERMS WILL BE LOCATED EVERY',I4,
     & ' TIME STEPS.')
      DO 37 K=1,9
      IFLAG(K)=0
   37 CONTINUE
      KONT=0
      JFLG=1
      Q=0
      ITCC=0
      ITPP=0
      DI2=2.D0*DI-1.D0
      KDX=1
C
C
C     XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
C     XXXXXXXXXX DON'T CHANGE ANYTHING ABOVE THIS LINE. XXXXXXXXXXXXXXXX
C
C       INITIALIZE SPECIFIC HEAT AND CONDUCTIVITY
C          IF TEMPERATURE DEPENDENT, INSERT EQUATIONS.
      DO 46 J=1,X
      DO 47 I=1,Y
      TEMP(I,J)=RAY(I,J,1)
      K=RAY(I,J,3)
      IF(RAY(I,J,1).LT.TPC(K)) GO TO 39
C            UNFROZEN
      CP(I,J)=1.160D0
      RO(I,J)=998.20D0
      GO TO 47
   39 CONTINUE
C                FROZEN
```

```
      CP(I,J)=.5815D0
      RO(I,J)=917.0D0
 47   CONTINUE
 46   CONTINUE
C  ********** *********** *********** ***********
2002 CONTINUE
C                   COMES HERE AFTER EVERY COMPLETE TIME STEP
C                   NOTE THAT Q=0 FOR 1ST ITERATION
C
C     IF SPECIFIED TEMPERATURES CHANGE WITH TIME, SPECIFY THE
C      CHANGES HERE. (NUMBERS 57 AND 58 ARE FREE FOR LOOPS IF NEEDED.)
C
C
C   ADJUST THERMAL PROPERTIES WITH TEMPERATURE.
C   IF APPROPRIATE, USE RAY(I,J,3) TO INDICATE MATERIAL TYPE.
C   (BUT DO NOT CHANGE THE NEXT 9 LINES.)
      DO 50 J=1,X
      DO 51 I=1,Y
      K=RAY(I,J,3)
      CPO(I,J)=CP(I,J)
      ROO(I,J)=RO(I,J)
      CC=TPC(K)+(TDEL/2.D0)
      DD=TPC(K)-(TDEL/2.D0)
      IF(RAY(I,J,1).LT.DD) GO TO 60
      IF(RAY(I,J,1).GT.CC) GO TO 61
C                UNDERGOING PHASE CHANGE
      ISTAT(I,J)=3
      TK(I,J)=2.210D0
      GO TO 51
 60   CONTINUE
C                FROZEN
      ISTAT(I,J)=1
      TK(I,J)=2.210D0
      RO(I,J)=917.0D0
      CP(I,J)=.58150D0
      GO TO 51
 61   CONTINUE
C                UNFROZEN
      ISTAT(I,J)=2
      TK(I,J)=.580D0
      RO(I,J)=998.2D0
      CP(I,J)=1.1630D0
 51   CONTINUE
 50   CONTINUE
C
C
C    XXXXXXXXXXXXXX DON'T CHANGE ANYTHING BELOW THIS LINE XXXXXXXXXXXXXXXXXXX
C    XXXXXXXXXXXXXX  UNLESS YOU KNOW WHAT YOU ARE DOING.  XXXXXXXXXXXXXXXXXXX
2001 CONTINUE
C                  COMES HERE AFTER EACH PASS
      IF(JFLG.EQ.1) GO TO 161
C              FOR 2ND PASS
      DO 1600 J=1,X
      KONT=0
      DO 41 K=1,Y
      EA(K)=0.0D0
      EB(K)=0.0D0
      C(K)=0.0D0
      D(K)=0.0D0
      Q(K)=0.0D0
 41   CONTINUE
      DO 1700 I=1,Y
      GO TO 1601
 161  CONTINUE
C              FOR 1ST PASS
      DO 160 I=1,Y
      KONT=0
      DO 42 K=1,X
      EA(K)=0.0D0
      EB(K)=0.0D0
      C(K)=0.0D0
      D(K)=0.0D0
      Q(K)=0.0D0
 42   CONTINUE
      DO 170 J=1,X
 1601  CONTINUE
C           FIGURE RESULTANT CONDUCTIVITIES BETWEEN NODES
      KONT=KONT+1
      INDEX=RAY(I,J,3)
      IF(I.EQ.1) GO TO 71
      TK1=(2.0D0*TK(I,J)*TK((I-1),J))/(TK(I,J)+TK((I-1),J))
 71   CONTINUE
      IF(J.EQ.1) GO TO 72
      TK2=(2.0D0*TK(I,J)*TK(I,(J-1)))/(TK(I,J)+TK(I,(J-1)))
 72   CONTINUE
      IF(I.EQ.Y) GO TO 73
      TK3=(2.0D0*TK(I,J)*TK((I+1),J))/(TK(I,J)+TK((I+1),J))
 73   CONTINUE
      IF(J.EQ.X) GO TO 74
      TK4=(2.0D0*TK(I,J)*TK(I,(J+1)))/(TK(I,J)+TK(I,(J+1)))
 74   CONTINUE
C           FIGURE  R3 FOR TIME T-DELT/2, R4 FOR TIME T
      IF(ISTAT(I,J).NE.3) GO TO 75
C           NODE PRESENTLY PHASE CHANGE
```

```
      R3=2.DO*DS*DS*ROPC(INDEX)*CPPC(INDEX)/DELT
      R4=R3
      GO TO 78
 75   CONTINUE
C           NODE NOT PHASE CHANGE PRESENTLY
      R3=2.DO*DS*DS*ROO(I,J)*CPO(I,J)/DELT
      R4=2.DO*DS*DS*RO(I,J)*CP(I,J)/DELT
 78   CONTINUE
C           ROUTE CORNERS
      IF(RAY(I,J,2).GE.5) GO TO 38
      IJ2=RAY(I,J,2)
      GO TO (101,102,103,104),IJ2
C           ROUTE THE REST OF THE NODES
 38   CONTINUE
      IJ2=RAY(I,J,2)-4
      GO TO (105,106,107,108,109,110,111,112,113,114,115,
     & 116,117,118,119,120,121,122,123,124,125,126,127,
     & 128,129,130,131,132),IJ2
C
 110  CONTINUE
C SEMI-INFINITE BOUNDARY
C           RIGHT SIDE
      IF(J.NE.X) GO TO 1100
      IFLAG(7)=11
      IF(JFLG.NE.1) GO TO 1108
      EA(KONT)=TK2/DI
      EB(KONT)=-TK2/DI-DI2*R4
      C(KONT)=0.DO
      D(KONT)=-TK1*DI2*RAY((I-1),J,1)-TK3*DI2*RAY((I+1),J,1)-
     & (TK(I,J)/DI)*TMPR(I)+(TK1*DI2+TK3*DI2-DI2*R3+TK(I,J)/DI)
     & *RAY(I,J,1)
      GO TO 165
 1108 CONTINUE
      EA(KONT)=TK1*DI2
      EB(KONT)=-TK1*DI2-TK3*DI2-DI2*R4
      C(KONT)=TK3*DI2
      D(KONT)=-(TK2/DI)*RAY(I,(J-1),1)-(TK(I,J)/DI)*TMPR(I)+
     & (TK2/DI+TK(I,J)/DI-DI2*R3)*RAY(I,J,1)
      GO TO 165
 1100  CONTINUE
C          LEFT SIDE
      IF(J.NE.1) GO TO 1101
      IFLAG(3)=11
      IF(JFLG.NE.1) GO TO 1104
      EA(KONT)=0.DO
      EB(KONT)=-TK4/DI-DI2*R4
      C(KONT)=TK4/DI
      D(KONT)=-TK1*DI2*RAY((I-1),J,1)-TK3*DI2*RAY((I+1),J,1)-
     & (TK(I,J)/DI)*TMPL(I)+(TK1*DI2+TK3*DI2-DI2*R3+TK(I,J)/DI)
     & *RAY(I,J,1)
      GO TO 165
 1104 CONTINUE
      EA(KONT)=TK1*DI2
      EB(KONT)=-TK1*DI2-TK3*DI2-DI2*R4
      C(KONT)=TK3*DI2
      D(KONT)=-(TK4/DI)*RAY(I,(J+1),1)-(TK(I,J)/DI)*TMPL(I)+
     & (TK4/DI+TK(I,J)/DI-DI2*R3)*RAY(I,J,1)
      GO TO 165
 1101  CONTINUE
C          TOP OF GRID
      IF (I.NE.1) GO TO 1102
      IFLAG(1)=11
      WRITE(1,9997) I,J
      GO TO 9999
 1102  CONTINUE
C          BOTTOM OF GRID
      IFLAG(5)=11
      IF(JFLG.NE.1) GO TO 1106
      EA(KONT)=TK2*DI2
      EB(KONT)=-TK2*DI2-TK4*DI2-R4*DI2
      C(KONT)=TK4*DI2
      D(KONT)=-(TK1/DI)*RAY((I-1),J,1)-(TK(I,J)/DI)*TMPB(J)+
     & (TK1/DI+TK(I,J)/DI-R3*DI2)*RAY(I,J,1)
      GO TO 165
 1106 CONTINUE
      EA(KONT)=TK1/DI
      EB(KONT)=-TK1/DI-R4*DI2
      C(KONT)=0.DO
      D(KONT)=-TK2*DI2*RAY(I,(J-1),1)-TK4*DI2*RAY(I,(J+1),1)-
     & (TK(I,J)/DI)*TMPB(J)+(TK2*DI2+TK4*DI2-DI2*R3+TK(I,J)/DI)
     & *RAY(I,J,1)
      GO TO 165
 109  CONTINUE
C   CONVECTIVE SURFACE BOUNDARY
C              TOP CONVECTIVE
      IFLAG(1)=12
      IF(JFLG.NE.1) GO TO 1092
      EA(KONT)=.5DO*TK2
      EB(KONT)=-.5DO*TK2-.5DO*TK4-.5DO*R4
      C(KONT)=.5DO*TK4
      D(KONT)=-TK3*RAY((I+1),J,1)+(TK3+H(INDEX)*DS-.5DO*R3)*
     & RAY(I,J,1)-H(INDEX)*DS*TMPT(J)
      GO TO 165
 1092 CONTINUE
```

```
      EA(KONT)=0.D0
      EB(KONT)=-TK3-.5D0*R4
      C(KONT)=TK3
      D(KONT)=-.5D0*TK2*RAY(I,(J-1),1)-.5D0*TK4*RAY(I,(J+1),1)+
     & (.5D0*TK2+.5D0*TK4-.5D0*R3+H(INDEX)*DS)*RAY(I,J,1)-H(INDEX)*DS*
     & TMPT(J)
      GO TO 165
  132 CONTINUE
C              RIGHT SIDE CONVECTIVE
      IFLAG(7)=12
      IF(JFLG.NE.1) GO TO 1094
      EA(KONT)=TK2
      EB(KONT)=-TK2-.5D0*R4
      C(KONT)=0.D0
      D(KONT)=-.5D0*TK1*RAY((I-1),J,1)-.5D0*TK3*RAY((I+1),J,1)+
     & (.5D0*TK1+.5D0*TK3-.5D0*R3+H(INDEX)*DS)*RAY(I,J,1)-
     & H(INDEX)*DS*TMPR(I)
      GO TO 165
 1094  CONTINUE
      EA(KONT)=.5D0*TK1
      EB(KONT)=-.5D0*TK1-.5D0*TK3+.5D0*R4
      C(KONT)=.5D0*TK3
      D(KONT)=-TK2*RAY(I,(J-1),1)+(TK2-.5D0*R3+H(INDEX)*DS)*
     & RAY(I,J,1)-H(INDEX)*DS*TMPR(I)
      GO TO 165
  131 CONTINUE
C            BOTTOM CONVECTIVE
      IFLAG(5)=12
      IF(JFLG.NE.1) GO TO 1096
      EA(KONT)=.5D0*TK2
      EB(KONT)=-.5D0*TK2-.5D0*TK4-.5D0*R4
      C(KONT)=.5D0*TK4
      D(KONT)=-TK1*RAY((I-1),J,1)+(TK1+H(INDEX)*DS-.5D0*R3)*
     & RAY(I,J,1)-H(INDEX)*DS*TMPB(J)
      GO TO 165
 1096 CONTINUE
      EA(KONT)=TK1
      EB(KONT)=-TK1-.5D0*R4
      C(KONT)=0.D0
      D(KONT)=-.5D0*TK2*RAY(I,(J-1),1)-.5D0*TK4*RAY(I,(J+1),1)+
     & (.5D0*TK2+.5D0*TK4-.5D0*R3+H(INDEX)*DS)*RAY(I,J,1)-H(INDEX)*DS*
     & TMPB(J)
      GO TO 165
  130 CONTINUE
C            LEFT SIDE CONVECTIVE
      IFLAG(3)=12
      IF(JFLG.NE.1) GO TO 1097
      EA(KONT)=0.D0
      EB(KONT)=-TK4-.5D0*R4
      C(KONT)=TK4
      D(KONT)=-.5D0*TK1*RAY((I-1),J,1)-.5D0*TK3*RAY((I+1),J,1)+
     & (.5D0*TK1+.5D0*TK3-.5D0*R3+H(INDEX)*DS)*RAY(I,J,1)-
     & H(INDEX)*DS*TMPL(I)
      GO TO 165
 1097 CONTINUE
      EA(KONT)=.5D0*TK1
      EB(KONT)=-.5D0*TK1-.5D0*TK3+.5D0*R4
      C(KONT)=.5D0*TK3
      D(KONT)=-TK4*RAY(I,(J+1),1)+(TK4-.5D0*R3+H(INDEX)*DS)*
     & RAY(I,J,1)-H(INDEX)*DS*TMPL(I)
      GO TO 165
  107  CONTINUE
C CONSTANT TEMP BOUNDARY
      EB(KONT)=1
      D(KONT)=RAY(I,J,1)
      IF(I.NE.1) GO TO 1070
      IFLAG(1)=10
      GO TO 165
 1070   CONTINUE
      IF(J.NE.1) GO TO 1071
      IFLAG(3)=10
      GO TO 165
 1071   CONTINUE
      IF(I.NE.Y) GO TO 1072
      IFLAG(5)=10
      GO TO 165
 1072   CONTINUE
      IF(J.NE.X) GO TO 1073
      IFLAG(7)=10
      GO TO 165
 1073  CONTINUE
  106 CONTINUE
C CONSTANT INTERIOR NODE
      EB(KONT)=1
      D(KONT)=RAY(I,J,1)
      GO TO 165
  105  CONTINUE
C VARIABLE INTERIOR NODE
      IF(JFLG.NE.1) GO TO 1050
      EA(KONT)=TK2
      C(KONT)=TK4
      EB(KONT)=-TK2-TK4-R4
      D(KONT)=-TK1*RAY((I-1),J,1)-TK3*RAY((I+1),J,1)+
     & (TK1+TK3-R3)*RAY(I,J,1)
```

```
      GO TO 165
 1050   CONTINUE
      EA(KONT)=TK1
      C(KONT)=TK3
      EB(KONT)=-TK1-TK3-R4
      D(KONT)=-TK2*RAY(I,(J-1),1)-TK4*RAY(I,(J+1),1)+(TK2+TK4-R3)*
     & RAY(I,J,1)
      GO TO 165
 129  CONTINUE
C CONSTANT HEAT FLUX
C          RIGHT SIDE
      IFLAG(7)=13
      IF(JFLG.NE.1) GO TO 1301
      EA(KONT)=TK2
      EB(KONT)=-TK2-.5D0*R4
      D(KONT)=-.5D0*TK1*RAY((I-1),J,1)-.5D0*TK3*RAY((I+1),J,1)+
     &  (.5D0*TK1+.5D0*TK3-.5D0*R3)*RAY(I,J,1)-FLXR(I)*DS
      C(KONT)=0.D0
      GO TO 165
 1301  CONTINUE
      EA(KONT)=.5D0*TK1
      EB(KONT)=-.5D0*TK1-.5D0*TK3-.5D0*R4
      C(KONT)=.5D0*TK3
      D(KONT)=-TK2*RAY(I,(J-1),1)+(TK2-.5D0*R3)*RAY(I,J,1)
     & -FLXR(I)*DS
      GO TO 165
 127   CONTINUE
C          LEFT SIDE CONST FLUX
      IFLAG(3)=13
      IF(JFLG.NE.1) GO TO 1303
      EA(KONT)=0.D0
      EB(KONT)=-TK4-.5D0*R4
      C(KONT)=TK4
      D(KONT)=-.5D0*TK1*RAY((I-1),J,1)-.5D0*TK3*RAY((I+1),J,1)+
     & (.5D0*TK1+.5D0*TK3-.5D0*R3)*RAY(I,J,1)-FLXL(I)*DS
      GO TO 165
 1303   CONTINUE
      EA(KONT)=.5D0*TK1
      EB(KONT)=-.5D0*TK1-.5D0*TK3-.5D0*R4
      C(KONT)=.5D0*TK3
      D(KONT)=-TK4*RAY(I,(J+1),1)+(TK4-.5D0*R3)*RAY(I,J,1)-FLXL(I)*DS
      GO TO 165
 108   CONTINUE
C             TOP CONST FLUX
      IFLAG(1)=13
      IF(JFLG.NE.1) GO TO 1305
      EA(KONT)=.5D0*TK2
      EB(KONT)=-.5D0*TK2-.5D0*TK4-.5D0*R4
      C(KONT)=.5D0*TK4
      D(KONT)=-TK3*RAY((I+1),J,1)+(TK3-.5D0*R3)*RAY(I,J,1)-FLXT(J)*DS
      GO TO 165
 1305  CONTINUE
      EA(KONT)=0.D0
      EB(KONT)=-TK3-.5D0*R4
      C(KONT)=TK3
      D(KONT)=-.5D0*TK2*RAY(I,(J-1),1)-.5D0*TK4*RAY(I,(J+1),1)+
     & (.5D0*TK2+.5D0*TK4-.5D0*R3)*RAY(I,J,1)-FLXT(J)*DS
      GO TO 165
 128  CONTINUE
C          BOTTOM CONST FLUX
      IF(JFLG.NE.1) GO TO 1307
      IFLAG(5)=13
      EA(KONT)=.5D0*TK2
      EB(KONT)=-.5D0*TK2-.5D0*TK4-.5D0*R4
      C(KONT)=.5D0*TK4
      D(KONT)=-TK1*RAY((I-1),J,1)+(TK1-.5D0*R3)*RAY(I,J,1)
     & -FLXB(J)*DS
      GO TO 165
 1307  CONTINUE
      EA(KONT)=TK1
      EB(KONT)=-TK1-.5D0*R4
      C(KONT)=0.D0
      D(KONT)=-.5D0*TK2*RAY(I,(J-1),1)-.5D0*TK4*RAY(I,(J+1),1)+
     & (.5D0*TK2+.5D0*TK4-.5D0*R3)*RAY(I,J,1)
     & -FLXB(J)*DS
      GO TO 165
 111  CONTINUE
C          NODE ADJ TO LEFT SEMI-INF
      IF(JFLG.NE.1) GO TO 1110
      EA(KONT)=TK2/DI
      EB(KONT)=-TK4-TK2/DI-R4
      C(KONT)=TK4
      D(KONT)=-TK1*RAY((I-1),J,1)-TK3*RAY((I+1),J,1)+(TK1+
     & TK3-R3)*RAY(I,J,1)
      GO TO 165
 1110 CONTINUE
      EA(KONT)=TK1
      EB(KONT)=-TK1-TK3-R4
      C(KONT)=TK3
      D(KONT)=-TK4*RAY(I,(J+1),1)-(TK2/DI)*RAY(I,(J-1),1)+
     & (TK4+TK2/DI-R3)*RAY(I,J,1)
      GO TO 165
 112  CONTINUE
C         NODE ADJ TO BOTTOM SEMI-INF
```

```
      IF(JFLG.NE.1) GO TO 1120
      EA(KONT)=TK2
      EB(KONT)=-TK2-TK4-R4
      C(KONT)=TK4
      D(KONT)=-TK1*RAY((I-1),J,1)-(TK3/DI)*RAY((I+1),J,1)+
     & (TK1+TK3/DI-R3)*RAY(I,J,1)
      GO TO 165
 1120 CONTINUE
      EA(KONT)=TK1
      EB(KONT)=-TK1-TK3/DI-R4
      C(KONT)=TK3/DI
      D(KONT)=-TK2*RAY(I,(J-1),1)-TK4*RAY(I,(J+1),1)+(TK2+
     & TK4-R3)*RAY(I,J,1)
      GO TO 165
 113  CONTINUE
C     NODE ADJ. TO RIGHT SEMI-INF
      IF(JFLG.NE.1) GO TO 1130
      EA(KONT)=TK2
      EB(KONT)=-TK2-TK4/DI-R4
      C(KONT)=TK4/DI
      D(KONT)=-TK1*RAY((I-1),J,1)-TK3*RAY((I+1),J,1)+
     & (TK1+TK3-R3)*RAY(I,J,1)
      GO TO 165
 1130 CONTINUE
      EA(KONT)=TK1
      EB(KONT)=-TK1-TK3-R4
      C(KONT)=TK3
      D(KONT)=(-TK4/DI)*RAY(I,(J+1),1)-TK2*RAY(I,(J-1),1)+
     & (TK2+TK4/DI-R3)*RAY(I,J,1)
      GO TO 165
 115  CONTINUE
C        SQUARE NODE ADJ TO 2 SEMI-INF SIDES
      IF(J.NE.2) GO TO 1150
      IF(I.NE.2) GO TO 1151
C              TOP LEFT
      WRITE(1,9997) I,J
      GO TO 9999
 1151 CONTINUE
C              BOTTOM LEFT
      IF(JFLG.NE.1) GO TO 1152
      EA(KONT)=TK2/DI
      EB(KONT)=-TK2/DI-TK4-R4
      C(KONT)=TK4
      D(KONT)=-TK1*RAY((I-1),J,1)+(TK3/DI)*RAY((I+1),J,1)+(TK1+
     & TK3/DI-R3)*RAY(I,J,1)
      GO TO 165
 1152 CONTINUE
      EA(KONT)=TK1
      EB(KONT)=-TK1-TK3/DI-R4
      C(KONT)=TK3/DI
      D(KONT)=-(TK2/DI)*RAY(I,(J-1),1)-TK4*RAY(I,(J+1),1)+
     & (TK2/DI+TK4-R3)*RAY(I,J,1)
      GO TO 165
 1150 CONTINUE
      IF(I.NE.2) GO TO 1153
C              TOP RIGHT
      WRITE(1,9997) I,J
      GO TO 9999
 1153 CONTINUE
C             BOTTOM RIGHT
      IF(JFLG.NE.1) GO TO 1154
      EA(KONT)=TK2
      EB(KONT)=-TK2-TK4/DI-R4
      C(KONT)=TK4/DI
      D(KONT)=-TK1*RAY((I-1),J,1)+(TK3/DI)*RAY((I+1),J,1)+
     & (TK1+TK3/DI-R3)*RAY(I,J,1)
      GO TO 165
 1154 CONTINUE
      EA(KONT)=TK1
      EB(KONT)=-TK1-TK3/DI-R4
      C(KONT)=TK3/DI
      D(KONT)=-TK2*RAY(I,(J-1),1)-(TK4/DI)*RAY(I,(J+1),1)+
     & (TK2+TK4/DI-R3)*RAY(I,J,1)
      GO TO 165
 121  CONTINUE
C        RIGHT SIDE CONST FLUX ADJ. TO BOTOM SEMI-INF
      IF(JFLG.NE.1) GO TO 1210
      EA(KONT)=TK2
      EB(KONT)=-TK2-.5D0*R4
      C(KONT)=0.D0
      D(KONT)=-.5D0*TK1*RAY((I-1),J,1)-(TK3/(2.D0*DI))*RAY((I+1),J,1)-
     & FLXR(I)*DS+(.5D0*TK1+TK3/(2.D0*DI)-.5D0*R3)*RAY(I,J,1)
      GO TO 165
 1210 CONTINUE
      EA(KONT)=.5D0*TK1
      EB(KONT)=-.5D0*TK1-TK3/(2.D0*DI)-.5D0*R4
      C(KONT)=TK3/(2.D0*DI)
      D(KONT)=-TK2*RAY(I,(J-1),1)-FLXR(I)*DS+(TK2-.5D0*R3)*RAY(I,J,1)
      GO TO 165
 123  CONTINUE
C     LEFT SIDE CONST FLUX ADJ TO BOTTOM SEMI-INF
```

```
      IF(JFLG.NE.1) GO TO 1230
      EA(KONT)=0.D0
      EB(KONT)=-TK4-.5D0*R4
      C(KONT)=TK4
      D(KONT)=-.5D0*TK1*RAY((I-1),J,1)-(TK3/(2.D0*DI))*RAY((I+1),J,1)-
     & FLXL(I)*DS+(.5D0*TK1+TK3/(2.D0*DI)-.5D0*R3)*RAY(I,J,1)
      GO TO 165
 1230 CONTINUE
      EA(KONT)=.5D0*TK1
      EB(KONT)=-.5D0*TK1-TK3/(2.D0*DI)-.5D0*R4
      C(KONT)=TK3/(2.D0*DI)
      D(KONT)=-TK4*RAY(I,(J+1),1)-FLXL(I)*DS+(TK4-.5D0*R3)*RAY(I,J,1)
      GO TO 165
  116   CONTINUE
C         SEMI-INF NODE ABOVE SEMI-INF CORNER NODE
      IF(J.NE.1) GO TO 1165
C            BOTTOM LEFT
      IF(JFLG.NE.1) GO TO 1160
      EA(KONT)=0.0D0
      EB(KONT)=-TK(I,J)/DI-TK4/DI+R4*DI2
      C(KONT)=TK4/DI
      D(KONT)=-TK1*DI2*RAY((I-1),J,1)-(TK3*DI2/DI)*RAY((I+1),J,1)
     & -(TK(I,J)/DI)*TMPL(I)+(TK1*DI2+TK3*DI2/DI-R3*DI2)*RAY(I,J,1)
      GO TO 165
 1160 CONTINUE
      EA(KONT)=TK1*DI2
      EB(KONT)=-(TK1+TK3/DI-R4)*DI2
      C(KONT)=TK3*DI2/DI
      D(KONT)=-(TK(I,J)/DI)*TMPL(I)-(TK4/DI)*RAY(I,(J+1),1)+
     & (TK(I,J)/DI+TK4/DI-R3*DI2)*RAY(I,J,1)
      GO TO 165
 1165 CONTINUE
C            BOTTOM RIGHT
      IF(JFLG.NE.1) GO TO 1166
      EA(KONT)=TK2/DI
      EB(KONT)=-TK(I,J)/DI-TK2/DI+R4*DI2
      C(KONT)=0.0D0
      D(KONT)=-TK1*DI2*RAY((I-1),J,1)-(TK3*DI2/DI)*RAY((I+1),J,1)
     & -(TK(I,J)/DI)*TMPL(I)+(TK1*DI2+TK3*DI2/DI-R3*DI2)*RAY(I,J,1)
      GO TO 165
 1166 CONTINUE
      EA(KONT)=TK1*DI2
      EB(KONT)=-(TK1+TK3/DI-R4)*DI2
      C(KONT)=TK3*DI2/DI
      D(KONT)=-(TK(I,J)/DI)*TMPR(I)-(TK2/DI)*RAY(I,(J-1),1)+
     & (TK(I,J)/DI+TK2/DI-R3*DI2)*RAY(I,J,1)
      GO TO 165
  117 CONTINUE
C       SEMI-INF NODE TO LEFT OF SEMI-INF CORNER NODE
      IF(JFLG.NE.1) GO TO 1170
      EA(KONT)=TK2*DI2
      EB(KONT)=(-TK4/DI-TK2-R4)*DI2
      C(KONT)=TK4*DI2/DI
      D(KONT)=(-TK1/DI)*RAY((I-1),J,1)-(TK(I,J)/DI)*TMPB(J)
     & +(TK1/DI+TK(I,J)/DI-R3*DI2)*RAY(I,J,1)
      GO TO 165
 1170 CONTINUE
      EA(KONT)=TK1/DI
      EB(KONT)=-(TK1+TK(I,J))/DI-R4*DI2
      C(KONT)=0.0D0
      D(KONT)=(-TK4*DI2/DI)*RAY(I,(J+1),1)-TK2*DI2*RAY(I,(J-1),1)-
     & (TK(I,J)/DI)*TMPB(J)+(TK2/DI+TK4-R3)*DI2*RAY(I,J,1)
      GO TO 165
  119 CONTINUE
C       SEMI-INF NODE TO RIGHT OF SEMI-INF CORNER NODE
      IF(JFLG.NE.1) GO TO 1190
      EA(KONT)=TK2*DI2/DI
      EB(KONT)=(-TK2/DI-TK4-R4)*DI2
      C(KONT)=TK4*DI2
      D(KONT)=(-TK1/DI)*RAY((I-1),J,1)-(TK(I,J)/DI)*TMPB(J)+
     & (TK1/DI+TK(I,J)/DI-R3*DI2)*RAY(I,J,1)
      GO TO 165
 1190 CONTINUE
      EA(KONT)=TK1/DI
      EB(KONT)=-(TK1+TK(I,J))/DI-R4*DI2
      C(KONT)=0.0D0
      D(KONT)=(-TK2*DI2/DI)*RAY(I,(J-1),1)-TK4*DI2*RAY(I,(J+1),1)-
     & (TK(I,J)/DI)*TMPB(J)+(TK2/DI+TK4-R3)*DI2*RAY(I,J,1)
      GO TO 165
  120 CONTINUE
C       TOP CONST FLUX NODE ADJ TO RIGHT SEMI-INF
      IF(JFLG.NE.1) GO TO 1200
      EA(KONT)=.50D0*TK2
      EB(KONT)=-.50D0*(TK2+TK4/DI+R4)
      C(KONT)=TK4/(2.0D0*DI)
      D(KONT)=-TK3*RAY((I+1),J,1)-DS*FLXT(J)+(TK3-.50D0*R3)
     & *RAY(I,J,1)
      GO TO 165
 1200 CONTINUE
      EA(KONT)=0.0D0
      EB(KONT)=-TK3-.50D0*R4
      C(KONT)=TK3
      D(KONT)=-1.*(TK4/(2.0D0*DI))*RAY(I,(J+1),1)-.5*TK2
```

```
      & *RAY(I,(J-1),1)+(TK4/(2.0D0*DI)+.50D0*TK2-.50D0*R3)
      & *RAY(I,J,1)-DS*FLXT(J)
  122 CONTINUE
C          TOP CONST FLUX ADJAC TO LEFT SEMI-INF
      IF(JFLG.NE.1) GO TO 1220
      EA(KONT)=TK2/(2.0D0*DI)
      EB(KONT)=-.50D0*(TK2/DI+TK4+R4)
      C(KONT)=.50D0*TK4
      D(KONT)=-TK3*RAY((I+1),J,1)-DS*FLXT(J)+
     & (TK3-.50D0*R3)*RAY(I,J,1)
      GO TO 165
 1220  CONTINUE
      EA(KONT)=0.0D0
      EB(KONT)=-TK3-.5*R4
      C(KONT)=TK3
      D(KONT)=-(TK2/(2.0D0*DI))*RAY(I,(J-1),1)-.50D0*TK4*
     & RAY(I,(J+1),1)+(TK2/(2.0D0*DI)+.50D0*TK4-.50D0*R3)
     & *RAY(I,J,1)-DS*FLXT(J)
      GO TO 165
  114 CONTINUE
  118 CONTINUE
  124 CONTINUE
  125 CONTINUE
  126 CONTINUE
      WRITE(1,9997) I,J
      GO TO 9999
C  ((((((((((((((((((((((((((((
C              EQUATIONS FOR THE CORNERS OF THE GRID
  101 CONTINUE
C           TOP LEFT-HAND CORNER
      GO TO (201,202,203,204,205,206,207,208,209,210),KRNR(1)
  202 CONTINUE
C             CONSTANT FLUX ON BOTH SIDES
      IFLAG(2)=1
      IF(JFLG.NE.1) GO TO 2020
      EA(KONT)=0.D0
      EB(KONT)=-.5D0*TK4-.25D0*R4
      C(KONT)=.5D0*TK4
      D(KONT)=-.5D0*TK3*RAY((I+1),J,1)+(.5D0*TK3-.25D0*R3)*RAY(I,J,1)-
     & .5D0*DS*(FLXT(J)+FLXL(I))
      GO TO 165
 2020  CONTINUE
      EA(KONT)=0.D0
      EB(KONT)=-.5D0*TK3-.25D0*R4
      C(KONT)=.5D0*TK3
      D(KONT)=-.5D0*TK4*RAY(I,(J+1),1)+(.5D0*TK4-.25D0*R3)*
     & RAY(I,J,1)-.5D0*DS*(FLXT(J)+FLXL(I))
      GO TO 165
  203 CONTINUE
C             CONVECTIVE ON BOTH SIDES
      IFLAG(2)=2
      IF(JFLG.NE.1) GO TO 2030
      EA(KONT)=0.D0
      EB(KONT)=-.5D0*TK4-.25D0*R4
      C(KONT)=.5D0*TK4
      D(KONT)=-.5D0*TK3*RAY((I+1),J,1)+(.5D0*TK3+H(INDEX)*DS-.25D0*R3)*
     & RAY(I,J,1)-.5D0*H(INDEX)*TMPT(J)-.5D0*H(INDEX)*TMPL(I)
      GO TO 165
 2030 CONTINUE
      EA(KONT)=0.D0
      EB(KONT)=-.5D0*TK3-.25D0*R4
      C(KONT)=.5D0*TK3
      D(KONT)=-.5D0*TK4*RAY(I,(J+1),1)+(.5D0*TK4+H(INDEX)*DS-.25D0*R3)*
     & RAY(I,J,1)-.5D0*H(INDEX)*(TMPT(J)+TMPL(I))
      GO TO 165
  204 CONTINUE
C             SEMI-INF ON BOTH SIDES
      IFLAG(2)=3
      WRITE(1,9997) I,J
      GO TO 9999
  206 CONTINUE
C               VERT CONST FLUX--HORIZ CONVECT
      IFLAG(2)=4
      IF(JFLG.NE.1) GO TO 2060
      EA(KONT)=0.0D0
      EB(KONT)=-.50D0*TK4-.250D0*R4
      C(KONT)=.50D0*TK4
      D(KONT)=-.50D0*TK3*RAY((I+1),J,1)-.50D0*FLXL(I)*DS
     & -.50D0*H(INDEX)*DS*TMPT(J)+(.50D0*TK3+.50D0*H(INDEX)*DS
     & -.250D0*R3)*RAY(I,J,1)
      GO TO 165
 2060 CONTINUE
      EA(KONT)=0.0D0
      EB(KONT)=-.50D0*TK3-.50D0*H(INDEX)*DS
      C(KONT)=.50D0*TK3
      D(KONT)=-.50D0*TK4*RAY(I,(J+1),1)-.50D0*FLXL(I)*DS
     & -.50D0*H(INDEX)*DS*TMPT(J)+(.50D0*TK4-.250D0*R3)
     & *RAY(I,J,1)
      GO TO 165
  205 CONTINUE
C               VERT CONVECT--HORIZ CONST FLUX
      IFLAG(2)=7
      IF(JFLG.NE.1) GO TO 2050
      EA(KONT)=0.0D0
```

```
      EB(KONT)=-.50D0*TK4-.50D0*H(INDEX)*DS-.250D0*R4
      C(KONT)=.50D0*TK4
      D(KONT)=-.50D0*FLXT(J)*DS-.50D0*TK3*RAY((I+1),J,1)+
     & (.50D0*TK3-.250D0*R3)*RAY(I,J,1)-.50D0*H(INDEX)*DS*TMPL(I)
      GO TO 165
 2050 CONTINUE
      EA(KONT)=0.0D0
      EB(KONT)=-.50D0*TK3-.250D0*R4
      C(KONT)=.50D0*TK3
      D(KONT)=-.50D0*TK4*RAY(I,(J+1),1)-.50D0*H(INDEX)*DS*TMPL(I)
     & +(.50D0*TK4+.50D0*H(INDEX)*DS-.50D0*FLXT(J)*DS-.250D0*R3)*
     & RAY(I,J,1)
      GO TO 165

  208 CONTINUE
C        VERT=SEMI-INF,HORIZ=CONST FLUX
      IF(JFLG.NE.1) GO TO 2080
      EA(KONT)=0.0D0
      EB(KONT)=-(TK(I,J)/(2.0D0*DI)+TK4/(2.0D0*DI)
     & +.50D0*DI2*R4)
      C(KONT)=TK4/(2.0D0*DI)
      D(KONT)=-FLXT(J)*DI2*DS-TK3*DI2*RAY((I+1),J,1)+
     & (TK3*DI2-.50D0*DI2*R3)*RAY(I,J,1)-(TK(I,J)/(2.0D0*DI))*TMPL(I)
      GO TO 165
 2080 CONTINUE
      EA(KONT)=0.0D0
      EB(KONT)=-(TK3+.50D0*R4)*DI2
      C(KONT)=TK3*DI2
      D(KONT)=-FLXT(J)*DI2*DS-(TK(I,J)/(2.0D0*DI))*TMPL(I)
     & -(TK4/(2.0D0*DI))*RAY(I,(J+1),1)+(TK(I,J)/(2.0D0*DI)+
     & TK4/(2.0D0*DI)-.50D0*DI2*R3)*RAY(I,J,1)
      GO TO 165
  210 CONTINUE
C          VERT=SEMI-INF,HORIZ=CONVECT
      IFLAG(2)=8
      IF(JFLG.NE.1) GO TO 2100
      EA(KONT)=0.0D0
      EB(KONT)=-(TK4+TK(I,J))/(2.0D0*DI)-.50D0*DI2*R4
      C(KONT)=TK4/(2.0D0*DI)
      D(KONT)=-H(INDEX)*DI2*TMPT(J)-(TK(I,J)/(2.0D0*DI))*TMPL(I)
     & -TK3*DI2*RAY((I+1),J,1)+(TK3*DI2+H(INDEX)*DI2-.50D0*DI2*R3)
     & *RAY(I,J,1)
      GO TO 165
 2100 CONTINUE
      EA(KONT)=0.0D0
      EB(KONT)=-TK3*DI2-H(INDEX)*DI2-.50D0*DI2*R4
      C(KONT)=TK3*DI2
      D(KONT)=-H(INDEX)*DI2*TMPT(J)-(TK4/(2.0D0*DI))*RAY(I,(J+1),1)
     & -(TK(I,J)/(2.0D0*DI))*TMPL(I)+((TK4+TK(I,J))/(2.0D0*DI)-
     & .50D0*DI2*R3)*RAY(I,J,1)
      GO TO 165
  207 CONTINUE
  209 CONTINUE
      WRITE(1,9997) I,J
      GO TO 9999
  201 CONTINUE
C        CONSTANT CORNER NODE
      EB(KONT)=1
      D(KONT)=RAY(I,J,1)
      IFLAG(2)=10
      GO TO 165
  102 CONTINUE
C          BOTTOM LEFT-HAND CORNER
      GO TO (211,212,213,214,215,216,217,218,219,220),KRNR(2)
  212 CONTINUE
C          CONST FLUX ON BOTH SIDES
      IFLAG(4)=1
      IF(JFLG.NE.1) GO TO 2120
      EA(KONT)=0.D0
      EB(KONT)=-.5D0*TK4-.25D0*R4
      C(KONT)=.5D0*TK4
      D(KONT)=-.5D0*TK1*RAY((I-1),J,1)+(.5D0*TK1-.25D0*R3)*RAY(I,J,1)
     & -.5*DS*(FLXB(J)+FLXL(I))
      GO TO 165
 2120  CONTINUE
      EA(KONT)=.5D0*TK1
      EB(KONT)=-.5D0*TK1-.25D0*R4
      C(KONT)=0.D0
      D(KONT)=-.5D0*TK4*RAY(I,(J+1),1)+(.5D0*TK4-.25D0*R3)*RAY(I,J,1)
     & -.5*DS*(FLXB(J)+FLXL(I))
      GO TO 165
  213 CONTINUE
C          CONVECT ON BOTH SIDES
      IFLAG(4)=2
      IF(JFLG.NE.1) GO TO 2130
      EA(KONT)=0.D0
      EB(KONT)=-.5D0*TK4-.25D0*R4
      C(KONT)=.5D0*TK4
      D(KONT)=-.5D0*TK1*RAY((I-1),J,1)+(.5D0*TK1+H(INDEX)*DS-.25D0*R3)*
     & RAY(I,J,1)-.5D0*H(INDEX)*(TMPL(I)+TMPB(J))
      GO TO 165
 2130 CONTINUE
      EA(KONT)=.5D0*TK1
      EB(KONT)=-.5D0*TK1-.25D0*R4
```

```
      C(KONT)=0.D0
      D(KONT)=-.5D0*TK4*RAY(I,(J+1),1)+(.5D0*TK4+H(INDEX)*DS-.25D0*R3)*
     & RAY(I,J,1)-.5D0*H(INDEX)*(TMPL(I)+TMPB(J))
      GO TO 165
  214 CONTINUE
C          SEMI-INF ON BOTH SIDES
      IFLAG(4)=3
      DINF=DI2/DI
      IF(JFLG.NE.1) GO TO 2140
      EA(KONT)=0.D0
      EB(KONT)=-TK4*DINF-(DI2*DI2)*R4
      C(KONT)=TK4*DINF
      D(KONT)=-TK1*DINF*RAY((I-1),J,1)-TK(I,J)*(DINF)*
     & TMPL(I)-TK(I,J)*DINF*TMPB(J)+(TK2*DINF
     & +2.D0*TK(I,J)*DINF-(DI2*DI2)*R3)*RAY(I,J,1)
      GO TO 165
 2140 CONTINUE
      EA(KONT)=TK1*DINF
      EB(KONT)=-TK1*DINF-(DI2*DI2)*R4
      C(KONT)=0.D0
      D(KONT)=-TK(I,J)*DINF*TMPL(I)-TK4*(DINF)*
     & RAY(I,(J+1),1)+(2.D0*TK(I,J)*DINF+TK4*DINF-
     & (DI2*DI2)*R3)*RAY(I,J,1)-(TK(I,J)*DINF)*TMPB(J)
      GO TO 165
  216 CONTINUE
C          VERT CONST FLUX--HORIZ CONVECT
      IFLAG(4)=4
      IF(JFLG.NE.1) GO TO 2160
      EA(KONT)=0.D00
      EB(KONT)=-.50D0*TK4-.250D0*R4
      C(KONT)=.50D0*TK4
      D(KONT)=-.50D0*TK1*RAY((I-1),J,1)-.50D0*FLXL(I)*DS-.50D0*
     & H(INDEX)*DS*TMPB(J)+(.50D0*TK3+.50D0*H(INDEX)*DS-.250D0*R3)*
     & RAY(I,J,1)
      GO TO 165
 2160 CONTINUE
      EA(KONT)=.50D0*TK1
      EB(KONT)=-.50D0*TK1-.50D0*H(INDEX)*DS
      C(KONT)=0.D00
      D(KONT)=-.50D0*TK4*RAY(I,(J+1),1)-.50D0*FLXL(I)*DS-.50D0*H(INDEX)*
     & DS*TMPB(J)+(.50D0*TK4-.250D0*R3)*RAY(I,J,1)
      GO TO 165
  215 CONTINUE
C     VERT CONVECT--HORIZ CONST FLUX
      IFLAG(4)=7
      IF(JFLG.NE.1) GO TO 2150
      EA(KONT)=0.D00
      EB(KONT)=-.50D0*TK4-.50D0*H(INDEX)*DS-.250D0*R4
      C(KONT)=.50D0*TK4
      D(KONT)=-.250D0*H(INDEX)*DS*TMPL(I)-.50D0*FLXB(J)*DS-.50D0*TK1
     & *RAY((I-1),J,1)+(.50D0*TK1-.250D0*R3)*RAY(I,J,1)
      GO TO 165
 2150 CONTINUE
      EA(KONT)=.50D0*TK1
      EB(KONT)=-.50D0*TK1-.250D0*R4
      C(KONT)=0.D00
      D(KONT)=-.50D0*TK4*RAY(I,(J+1),1)-.50D0*H(INDEX)*DS*TMPL(I)
     & +(.50D0*TK4+.50D0*H(INDEX)*DS-.50D0*FLXB(J)*DS-.250D0*R3)
     & *RAY(I,J,1)
      GO TO 165
  217 CONTINUE
C      VERT CONST FLUX--HORIZ SEMI-INF
      IFLAG(4)=5
      IF(JFLG.NE.1) GO TO 2170
      EA(KONT)=0.D0
      EB(KONT)=-TK4*DI2-.25D0*DI2*R4
      C(KONT)=TK4*DI2
      D(KONT)=-(TK1/(2.D0*DI))*RAY((I-1),J,1)-(TK(I,J)/(2.D0*DI))*
     & TMPB(J)-FLXL(I)*DI2*DS+(TK1/(2.D0*DI)+TK(I,J)/(2.D0*DI)-
     & .25D0*DI2*R3)*RAY(I,J,1)
      GO TO 165
 2170 CONTINUE
      EA(KONT)=TK1/(2.D0*DI)
      EB(KONT)=-TK1/(2.D0*DI)-TK(I,J)/(2.D0*DI)-.25D0*DI2*R4
      C(KONT)=0.D0
      D(KONT)=-TK4*DI2*RAY(I,(J+1),1)-(TK(I,J)/(2.D0*DI))*TMPB(J)-
     & FLXL(I)*DI2*DS+(TK4*DI2-.25D0*DI2*R3)*RAY(I,J,1)
      GO TO 165
  218 CONTINUE
C             VERT=SEMI-INF,HORIZ=CONST FLUX
      IFLAG(4)=9
      IF(JFLG.NE.1) GO TO 2180
      EA(KONT)=0.D00
      EB(KONT)=-(TK4+TK(I,J))/(2.D00*DI)-.50D0*DI2*R4
      C(KONT)=TK4/(2.D00*DI)
      D(KONT)=-FLXB(J)*DI2*DS-(TK(I,J)/(2.D00*DI))*TMPL(I)-TK1*DI2
     & *RAY((I-1),J,1)+(TK1*DI2-.50D0*DI2*R3)*RAY(I,J,1)
      GO TO 165
 2180 CONTINUE
      EA(KONT)=TK1*DI2
      EB(KONT)=-TK1*DI2-.50D0*DI2*R4
      C(KONT)=0.D00
      D(KONT)=-(TK4/(2.D00*DI))*RAY(I,(J+1),1)-(TK(I,J)/(2.D00*DI))*
```

```
     & TMPL(I)-FLXB(J)*DI2*DS+((TK4+TK(I,J))/(2.0D0*DI)-.50D0*DI2*R3)
     & *RAY(I,J,1)
      GO TO 165
  219 CONTINUE
C          VERT=CONVECT,HORIZ=SEMI-INF
      IFLAG(4)=6
      IF(JFLG.NE.1) GO TO 2190
      EA(KONT)=0.0D0
      EB(KONT)=-TK4*DI2-H(INDEX)*DI2-.50D0*DI2*R4
      C(KONT)=TK4*DI2
      D(KONT)=-H(INDEX)*DI2*TMPL(I)-(TK1/(2.0D0*DI))*RAY((I-1),J,1)
     & -(TK(I,J)/(2.0D0*DI))*TMPB(J)+((TK1+TK(I,J))/(2.0D0*DI)
     & -.50D0*DI2*R3)*RAY(I,J,1)
      GO TO 165
 2190 CONTINUE
      EA(KONT)=TK1/(2.0D0*DI)
      EB(KONT)=-(TK1+TK(I,J))/(2.0D0*DI)-.50D0*DI2*R4
      C(KONT)=0.0D0
      D(KONT)=-H(INDEX)*DI2*TMPL(I)-(TK(I,J)/(2.0D0*DI))*TMPB(J)
     & -TK4*DI2*RAY(I,(J+1),1)+(TK4*DI2+H(INDEX)*DI2-.50D0*DI2*R3)
     & *RAY(I,J,1)
      GO TO 165
  220 CONTINUE
C          VERT=SEMI-INF,HORIZ=CONVECT
      IFLAG(4)=8
      IF(JFLG.NE.1) GO TO 2200
      EA(KONT)=0.0D0
      EB(KONT)=-(TK4+TK(I,J))/(2.0D0*DI)-.50D0*DI2*R4
      C(KONT)=TK4/(2.0D0*DI)
      D(KONT)=-H(INDEX)*DI2*TMPB(J)-(TK(I,J)/(2.0D0*DI))*TMPL(I)
     & -TK1*DI2*RAY((I-1),J,1)+(TK1*DI2+H(INDEX)*DI2-.50D0*DI2*R3)
     & *RAY(I,J,1)
      GO TO 165
 2200 CONTINUE
      EA(KONT)=TK1*DI2
      EB(KONT)=-TK1*DI2-H(INDEX)*DI2-.50D0*DI2*R4
      C(KONT)=0.0D0
      D(KONT)=-H(INDEX)*DI2*TMPB(J)-(TK4/(2.0D0*DI))*RAY(I,(J+1),1)
     & -(TK(I,J)/(2.0D0*DI))*TMPL(I)+((TK4+TK(I,J))/(2.0D0*DI)
     & -.50D0*DI2*R3)*RAY(I,J,1)
      GO TO 165
  211 CONTINUE
C          CONST CORNER NODE
      EB(KONT)=1
      D(KONT)=RAY(I,J,1)
      IFLAG(4)=10
      GO TO 165
  103 CONTINUE
C        BOTTOM RIGHT-HAND CORNER
      GO TO (221,222,223,224,225,226,227,228,229,230),KRNR(3)
  222 CONTINUE
C         CONST FLUX ON BOTH SIDES
      IF(JFLG.NE.1) GO TO 2220
      EA(KONT)=.5D0*TK2
      EB(KONT)=-.5D0*TK2-.25D0*R4
      C(KONT)=0.D0
      D(KONT)=-.5D0*TK1*RAY((I-1),J,1)+(.5D0*TK1-.25D0*R3)*RAY(I,J,1)
     & -.5D0*DS*(FLXB(J)+FLXR(I))
      GO TO 165
 2220  CONTINUE
      EA(KONT)=.5D0*TK1
      EB(KONT)=-.5D0*TK1-.25D0*R4
      C(KONT)=0.D0
      D(KONT)=-.5D0*TK2*RAY(I,(J-1),1)+(.5D0*TK2-.25D0*R3)*RAY(I,J,1)
     & -.5D0*DS*(FLXB(J)+FLXR(I))
      IFLAG(6)=1
      GO TO 165
  223 CONTINUE
C         CONVECTIVE ON BOTH SIDES
      IFLAG(6)=2
      IF(JFLG.NE.1) GO TO 2230
      EA(KONT)=.5D0*TK2
      EB(KONT)=-.5D0*TK2-.25D0*R4
      C(KONT)=0.D0
      D(KONT)=-.5D0*TK1*RAY((I-1),J,1)+(.5D0*TK1+H(INDEX)*DS-.25D0*R3)*
     & RAY(I,J,1)-.5D0*H(INDEX)*(TMPB(J)+TMPR(I))
      GO TO 165
 2230 CONTINUE
      EA(KONT)=.5D0*TK1
      EB(KONT)=-.5D0*TK1-.25D0*R4
      D(KONT)=0.D0
      D(KONT)=-.5D0*TK2*RAY(I,(J-1),1)+(.5D0*TK2+H(INDEX)*DS-.25D0*R3)*
     & RAY(I,J,1)-.5D0*H(INDEX)*(TMPR(I)+TMPB(J))
      GO TO 165
  224 CONTINUE
C         SEMI-INF ON BOTH SIDES
      IFLAG(6)=3
      DINF=DI2/DI
      IF(JFLG.NE.1) GO TO 2240
      EA(KONT)=TK2*DINF
      EB(KONT)=-TK2*DINF-(DI2*DI2)*R4
      C(KONT)=0.D0
      D(KONT)=-TK1*DINF*RAY((I-1),J,1)-TK(I,J)*DINF*
```

```
     & TMPR(I)-TK(I,J)*DINF*TMPB(J)+(TK1*DINF+
     & 2.D0*TK(I,J)*DINF-(DI2*DI2)*R3)*RAY(I,J,1)
      GO TO 165
 2240 CONTINUE
      EA(KONT)=TK1*DINF
      EB(KONT)=-TK1*DINF-(DI2*DI2)*R4
      C(KONT)=C.D0
      D(KONT)=-TK(I,J)*DINF*TMPR(I)-TK2*DINF*
     & RAY(I,(J-1),1)+(2.D0*TK(I,J)*DINF+TK2*DINF-
     & DI2*DI2*R3)*RAY(I,J,1)-(TK(I,J)*DINF)*TMPB(J)
      GO TO 165
  226 CONTINUE
C           VERT CONST FLUX-- HORIZ CONVECT
      IFLAG(6)=4
      IF(JFLG.NE.1) GO TO 2260
      EA(KONT)=.50D0*TK2
      EB(KONT)=-.50D0*TK2-.250D0*R4
      C(KONT)=0.0D0
      D(KONT)=-.50D0*TK1*RAY((I-1),J,1)-.50D0*FLXR(I)*DS-.50D0*H(INDEX)
     & *DS*TMPB(J)+(.50D0*TK1+.50D0*H(INDEX)*DS-.250D0*R3)*RAY(I,J,1)
      GO TO 165
 2260 CONTINUE
      EA(KONT)=.50D0*TK1
      EB(KONT)=-.50D0*TK1-.50D0*H(INDEX)*DS
      C(KONT)=0.0D0
      D(KONT)=-.50D0*TK2*RAY(I,(J-1),1)-.50D0*FLXR(I)*DS-.50D0*H(INDEX)
     & *DS*TMPB(J)+(.50D0*TK2-.250D0*R3)*RAY(I,J,1)
      GO TO 165
  225 CONTINUE
C           VERT CONVECT-- HORIZ CONST FLUX
      IFLAG(6)=7
      IF(JFLG.NE.1) GO TO 2250
      EA(KONT)=.50D0*TK2
      EB(KONT)=-.50D0*TK2-.50D0*H(INDEX)*DS-.250D0*R4
      C(KONT)=0.0D0
      D(KONT)=-.50D0*H(INDEX)*DS*TMPR(I)-.50D0*FLXB(J)*DS-.50D0*TK1
     & *RAY((I-1),J,1)+(.50D0*TK1-.250D0*R3)*RAY(I,J,1)
      GO TO 165
 2250 CONTINUE
      EA(KONT)=.50D0*TK1
      EB(KONT)=-.50D0*TK1-.250D0*R4
      C(KONT)=0.0D0
      D(KONT)=-.50D0*TK2*RAY(I,(J-1),1)-.50D0*H(INDEX)*DS*TMPR(I)+
     & (.50D0*TK2+.50D0*H(INDEX)*DS-.50D0*FLXB(J)*DS-.250D0*R3)
     & *RAY(I,J,1)
      GO TO 165
  227 CONTINUE
C     VERT CONST FLUX, HORIZ SEMI-INF
      IFLAG(6)=5
      IF(JFLG.NE.1) GO TO 2270
      EA(KONT)=TK2*DI2
      EB(KONT)=-TK2*DI2-.25D0*DI2*R4
      C(KONT)=0.D0
      D(KONT)=-(TK1/(2.D0*DI))*RAY((I-1),J,1)-(TK(I,J)/(2.D0*DI))*
     & TMPB(J)-FLXR(I)*DI2*DS+(TK1/(2.D0*DI)+TK(I,J)/(2.D0*DI)-
     & .25D0*DI2*R3)*RAY(I,J,1)
      GO TO 165
 2270 CONTINUE
      EA(KONT)=TK1/(2.D0*DI)
      EB(KONT)=-TK1/(2.D0*DI)-TK(I,J)/(2.D0*DI)-.25D0*DI2*R4
      C(KONT)=0.D0
      D(KONT)=-TK2*DI2*RAY(I,(J-1),1)-(TK(I,J)/(2.D0*DI))*TMPB(J)-
     & FLXR(I)*DI2*DS+(TK2*DI2-.25D0*DI2*R3)*RAY(I,J,1)
      GO TO 165
  228 CONTINUE
C           VERT=SEMI-INF,HORIZ=CONST FLUX
      IFLAG(6)=9
      IF(JFLG.NE.1) GO TO 2280
      EA(KONT)=TK2/(2.0D0*DI)
      EB(KONT)=-(TK2+TK(I,J))/(2.0D0*DI)-.50D0*DI2*R4
      C(KONT)=0.0D0
      D(KONT)=-FLXB(J)*DI2*DS-(TK(I,J)/(2.0D0*DI))*TMPR(I)
     & -TK1*DI2*RAY((I-1),J,1)+(TK1*DI2-.50D0*DI2*R3)*RAY(I,J,1)
      GO TO 165
 2280 CONTINUE
      EA(KONT)=TK1*DI2
      EB(KONT)=-TK1*DI2-.50D0*DI2*R4
      C(KONT)=0.0D0
      D(KONT)=-(TK2/(2.0D0*DI))*RAY(I,(J-1),1)-(TK(I,J)/(2.0D0*DI))*
     & TMPR(I)-FLXB(J)*DI2*DS+((TK2+TK(I,J))/(2.0D0*DI)-.50D0*DI2*R3)*
     & RAY(I,J,1)
      GO TO 165
  229 CONTINUE
C           VERT=CONVECT,HORIZ=SEMI-INF
      IFLAG(6)=6
      IF(JFLG.NE.1) GO TO 2290
      EA(KONT)=TK2*DI2
      EB(KONT)=-TK2*DI2-H(INDEX)*DI2-.50D0*DI2*R4
      C(KONT)=0.0D0
      D(KONT)=-H(INDEX)*DI2*TMPR(I)-(TK1/(2.0D0*DI))*RAY((I-1),J,1)
     & -(TK(I,J)/(2.0D0*DI))*TMPB(J)+((TK1+TK(I,J))/(2.0D0*DI)
     & -.50D0*DI2*R3)*RAY(I,J,1)
      GO TO 165
```

```
 2290 CONTINUE
      EA(KONT)=TK1/(2.0D0*DI)
      EB(KONT)=-(TK1+TK(I,J))/(2.0D0*DI)-.50D0*DI2*R4
      C(KONT)=0.0D0
      D(KONT)=-H(INDEX)*DI2*TMPR(I)-(TK(I,J)/(2.0D0*DI))*TMPB(J)
     & -TK2*DI2*RAY(I,(J-1),1)+(TK4*DI2+H(INDEX)*DI2-.50D0*DI2*R3)*
     & RAY(I,J,1)
      GO TO 165
  230 CONTINUE
C                VERT=SEMI-INF,HORIZ=CONVECT
      IFLAG(6)=8
      IF(JFLG.NE.1) GO TO 2300
      EA(KONT)=TK2/(2.0D0*DI)
      EB(KONT)=-(TK2+TK(I,J))/(2.D0*DI)-.50D0*DI2*R4
      C(KONT)=0.0D0
      D(KONT)=-H(INDEX)*DI2*TMPB(J)-(TK(I,J)/(2.0D0*DI))*TMPR(I)
     & -TK1*DI2*RAY((I-1),J,1)+(H(INDEX)*DI2+TK1*DI2-.50D0*DI2*R3)*
     & RAY(I,J,1)
      GO TO 165
 2300 CONTINUE
      EA(KONT)=TK1*DI2
      EB(KONT)=-TK1*DI2-.50D0*DI2*R4-H(INDEX)*DI2
      C(KONT)=0.0D0
      D(KONT)=-H(INDEX)*DI2*TMPB(J)-(TK2/(2.0D0*DI))*RAY(I,(J-1),1)
     & -(TK(I,J)/(2.0D0*DI))*TMPR(I)+((TK2+TK(I,J))/(2.0D0*DI)
     & -.50D0*DI2*R3)*RAY(I,J,1)
      GO TO 165
  221 CONTINUE
C       CONSTANT CORNER NODE
      EB(KONT)=1
      D(KONT)=RAY(I,J,1)
      IFLAG(6)=10
      GO TO 165
  104 CONTINUE
C                TOP RIGHT-HAND CORNER
      GO TO (231,232,233,234,235,236,237,238,239,240),KRNR(4)
  232 CONTINUE
C       CONST FLUX ON BOTH SIDES
      IF(JFLG.NE.1) GO TO 2320
      EA(KONT)=.5D0*TK2
      EB(KONT)=-.5D0*TK2-.25D0*R4
      C(KONT)=0.D0
      D(KONT)=-.5D0*TK3*RAY((I+1),J,1)+(.5D0*TK3-.25D0*R3)*
     & RAY(I,J,1)-.5D0*DS*(FLXT(J)+FLXR(I))
      GO TO 165
 2320  CONTINUE
      EA(KONT)=0.D0
      EB(KONT)=-.5D0*TK3-.25D0*R4
      C(KONT)=.5D0*TK3
      D(KONT)=-.5D0*TK2*RAY(I,(J-1),1)+(.5D0*TK2-.25D0*R3)*RAY(I,J,1)-
     & .5D0*DS*(FLXT(J)+FLXR(I))
      IFLAG(8)=1
      GO TO 165
  233 CONTINUE
C          CONVECTIVE ON BOTH SIDES
      IFLAG(8)=2
      IF(JFLG.NE.1) GO TO 2330
      EA(KONT)=.5D0*TK2
      EB(KONT)=-.5D0*TK2-.25D0*R4
      C(KONT)=0.D0
      D(KONT)=-.5D0*TK3*RAY((I+1),J,1)+(.5D0*TK3+H(INDEX)*DS-.25D0*R3)*
     & RAY(I,J,1)-.5D0*H(INDEX)*TMPT(J)-.5D0*H(INDEX)*TMPR(I)
      GO TO 165
 2330 CONTINUE
      EA(KONT)=0.D0
      EB(KONT)=-.5D0*TK3-.25D0*R4
      C(KONT)=.5D0*TK3
      D(KONT)=-.5D0*TK2*RAY(I,(J-1),1)+(.5D0*TK2+H(INDEX)*DS-.25D0*R3)*
     & RAY(I,J,1)-.5D0*H(INDEX)*TMPT(J)-.5D0*H(INDEX)*TMPR(I)
      GO TO 165
  234 CONTINUE
C            SEMI-INF ON BOTH SIDES
      IFLAG(8)=3
      WRITE(1,9997) I,J
      GO TO 9999
  236 CONTINUE
C              VERT CONST FLUX--HORIZ CONVECT
      IFLAG(8)=4
      IF(JFLG.NE.1) GO TO 2360
      EA(KONT)=.50D0*TK2
      EB(KONT)=-.50D0*TK2-.250D0*R4
      C(KONT)=0.0D0
      D(KONT)=-.50D0*TK3*RAY((I+1),J,1)-.50D0*FLXR(I)*DS
     & -.50D0*H(INDEX)*DS*TMPT(J)+(.50D0*TK3+.50D0*H(INDEX)*DS
     & -.250D0*R3)*RAY(I,J,1)
      GO TO 165
 2360 CONTINUE
      EA(KONT)=0.0D0
      EB(KONT)=-.50D0*TK3-.50D0*H(INDEX)*DS
      C(KONT)=.50D0*TK3
      D(KONT)=-.50D0*TK2*RAY(I,(J-1),1)-.50D0*FLXR(I)*DS
     & -.50D0*H(INDEX)*DS*TMPT(J)+(.50D0*TK2-.250D0*R3)*RAY(I,J,1)
      GO TO 165
C              VERT SIDE=SEMI-INF, HORIZ CONST FLUX
```

```
  238   CONTINUE
        IF(JFLG.NE.1) GO TO 2380
        EA(KONT)=TK2/(2.0D0*DI)
        EB(KONT)=-.50D0*(TK(I,J)/DI+TK2/DI+DI2*R4)
        C(KONT)=0.0D0
        D(KONT)=-TK3*DI2*RAY((I+1),J,1)-FLXT(J)*DI2*DS+
       & (TK3*DI2-.50D0*DI2*R3)*RAY(I,J,1)
        GO TO 165
  2380  CONTINUE
        EA(KONT)=0.0D0
        EB(KONT)=-TK3*DI2-.50D0*DI2*R4
        C(KONT)=TK3*DI2
        D(KONT)=(-TK(I,J)/(2.0D0*DI))*TMPR(I)-(TK2/(2.0D0*DI))*
       & RAY(I,(J+1),1)+(TK(I,J)/(2.0D0*DI)+TK2/(2.0D0*DI)-.50D0*DI2
       & *R3)*RAY(I,J,1)-FLXT(J)*DI2*DS
        GO TO 165
  235   CONTINUE
C            VERT CONVECT--HORIZ CONST FLUX
        IFLAG(8)=7
        IF(JFLG.NE.1) GO TO 2350
        EA(KONT)=.50D0*TK2
        EB(KONT)=-.50D0*TK2-.50D0*H(INDEX)*DS-.250D0*R4
        C(KONT)=0.0D0
        D(KONT)=-.50D0*FLXT(J)*DS-.50D0*TK3*RAY((I+1),J,1)+
       & (.50D0*TK3-.250D0*R3)*RAY(I,J,1)-.50D0*H(INDEX)*DS*TMPR(I)
        GO TO 165
  2350  CONTINUE
        EA(KONT)=0.0D0
        EB(KONT)=-.50D0*TK3-.250D0*R4
        C(KONT)=.50D0*TK3
        D(KONT)=-.50D0*TK2*RAY(I,(J-1),1)-.50D0*H(INDEX)*DS*
       & TMPR(I)+(.50D0*TK2+.50D0*H(INDEX)*DS-.50D0*FLXT(J)*DS
       & -.250D0*R3)*RAY(I,J,1)
        GO TO 165
  240   CONTINUE
C            VERT=SEMI-INF,HORIZ=CONVECT
        IFLAG(8)=8
        IF(JFLG.NE.1) GO TO 2400
        EA(KONT)=TK2/(2.0D0*DI)
        EB(KONT)=-(TK2+TK(I,J))/(2.0D0*DI)-.50D0*DI2*R4
        C(KONT)=0.0D0
        D(KONT)=-H(INDEX)*DI2*TMPT(J)-(TK(I,J)/(2.0D0*DI))*TMPR(I)
       & -TK3*DI2*RAY((I+1),J,1)+(H(INDEX)*DI2+TK3*DI2-.50D0*DI2*R3)*
       & RAY(I,J,1)
        GO TO 165
  2400  CONTINUE
        EA(KONT)=0.0D0
        EB(KONT)=-TK3*DI2-.50D0*DI2*R4-H(INDEX)*DI2
        C(KONT)=TK3*DI2
        D(KONT)=-H(INDEX)*DI2*TMPT(J)-(TK2/(2.0D0*DI))*RAY(I,(J-1),1)
       & -(TK(I,J)/(2.0D0*DI))*TMPR(I)+((TK2+TK(I,J))/(2.0D0*DI)-.50D0*
       & DI2*R3)*RAY(I,J,1)
        GO TO 165
  237   CONTINUE
  239   CONTINUE
        WRITE(1,9997) I,J
        GO TO 9999
  231   CONTINUE
C          CONSTANT CORNER NODE
        EB(KONT)=1
        D(KONT)=RAY(I,J,1)
        IFLAG(8)=10
        GO TO 165
C ((((((((((((((((((((((((((((((((


  165   CONTINUE
C                      CHECK WHICH PASS
        IF(JFLG.EQ.2) GO TO 1602
  170   CONTINUE
C                    A ROW SET UP ON 1ST PASS NOW.
        L=X
        IF=1
        CALL TRIDIG(L,EA,EB,C,D)
        DO 1607 K=1,X
        TEMP(I,K)=D(K)
  1607   CONTINUE
  160   CONTINUE
C                  THE WHOLE GRID 1ST PASS COMPUTED NOW.
        JFLG=2
        KONT=0
        DO 173 I=1,Y
        DO 174 J=1,X
        OLDT(I,J)=RAY(I,J,1)
        RAY(I,J,1)=TEMP(I,J)
  174   CONTINUE
  173   CONTINUE
        GO TO 2001
  1602   CONTINUE
  1700   CONTINUE
C            A COLUMN SET UP ON 2ND PASS NOW.
  1613  CONTINUE
        IF=1
        L=Y
```

```
      CALL TRIDIG(L,EA,EB,C,D)
      DO 1608 K=1,Y
      TEMP(K,J)=D(K)
 1608 CONTINUE
 1600 CONTINUE
C           WHOLE GRID 2ND PASS COMPLETE NOW.
      DO 1609 J=1,X
      DO 1610 I=1,Y
      RAY(I,J,1)=TEMP(I,J)
C         CHECK IF FRONT SKIPPED THE NODE
      K=RAY(I,J,3)
      CC=TPC(K)+(TDEL/2.D0)
      DD=TPC(K)-(TDEL/2.D0)
      IF((OLDT(I,J).GT.CC).AND.(TEMP(I,J).LT.DD)) GO TO 188
      IF((OLDT(I,J).LT.DD).AND.(TEMP(I,J).GT.CC)) GO TO 189
      GO TO 1610
  188 CONTINUE
C         SKIPPED FROM UNFROZEN TO FROZEN
      RAY(I,J,1)=CC+(CP(I,J)/CPPC(K))*(RAY(I,J,1)-CC)
      GO TO 1610
  189 CONTINUE
CC        SKIPPED FROM FROZEN TO UNFROZEN
      RAY(I,J,1)=DD+(CP(I,J)/CPPC(K))*(RAY(I,J,1)-DD)
 1610 CONTINUE
 1609 CONTINUE
      JFLG=1
      KONT=0
      Q=Q+1
      ITCC=ITCC+1
      ITPP=ITPP+1
C
C *************
C           WRITE OUTPUT
C    ITRT IS NO. OF TIME STEPS BEFORE TEMPERATURES PRINTED
C    ITCC COUNTS FROM 1 TO ITRT
C    ITPC IS NO. OF TIME STEPS BEFORE ISOTHM IS CALLED
C    ITPP COUNTS FROM 1 TO ITPC
C
      TT=Q*DELT
C
C            FIGURE & WRITE ISOTHERM LOCATIONS
      IF(ITPP.NE.ITPC) GO TO 920
      CALL ISOTHM
      ITPP=0
  920 CONTINUE
C
C            WRITE TEMPERATURE DISTRIBUTIONS
      IF(ITCC.NE.ITRT) GO TO 888
      WRITE(11,893) Q,TT
  893 FORMAT(/,/,/,1X,'TEMPERATURES AFTER',I4,' TIME STEPS (',
     & F12.3,' HRS):')
      RRX=X/17
      JJX=RRX
      JJR=X-(17*JJX)
      DO 531 JJ=1,JJX
      IF(JJX.EQ.0) GO TO 531
      J2=17*JJ
      J1=J2-16
      WRITE(11,500) ((RAY(I,J,1),J=J1,J2),I=1,Y)
      IF(JJ.LT.JJX) WRITE(11,503)
  531 CONTINUE
      J2=X
      J1=X-JJR+1
      IF(JJR.EQ.0) GO TO 532
      IF(JJX.NE.0) WRITE(11,503)
      DO 532 I=1,Y
      WRITE(11,500) (RAY(I,J,1),J=J1,J2)
  532 CONTINUE
      ITCC=0
  888 CONTINUE
C ******
C
C CHECK NO. OF WHOLE TIME STEPS AND GO BACK TO START IF NOT DONE
      IF(Q.GE.IMAX) GO TO 172
      JFLG=1
      KONT=0
      GO TO 2002
  172 CONTINUE
  200 CONTINUE
C    WRITE BOUNDARY TYPES INTO RESULT
      WRITE(6,404)
  404 FORMAT(/,2X,'BOUNDARY CONDITIONS:')
      DO 410 K=1,8
      KKK=K
      IF(IFLAG(K).EQ.0) GO TO 410
      GO TO (400,450,401,451,402,452,403,453),KKK
  400 WRITE(6,405)
  405 FORMAT(/,1X,'TOP BOUNDARY')
      GO TO 470
  401 WRITE(6,406)
  406 FORMAT(1X,'LEFT BOUNDARY')
      GO TO 470
  402 WRITE(6,407)
  407 FORMAT(1X,'BOTTOM BOUNDARY')
```

```
      GO TO 470
 403  WRITE(6,408)
 408  FORMAT(1X,'RIGHT BOUNDARY')
      GO TO 470
 450  WRITE(6,460)
 460  FORMAT(1X,'TOP LEFT CORNER')
      GO TO 470
 451  WRITE(6,461)
 461  FORMAT(1X,'BOTTOM LEFT CORNER')
      GO TO 470
 452  WRITE(6,462)
 462  FORMAT(1X,'BOTTOM RIGHT CORNER')
      GO TO 470
 453  WRITE(6,463)
 463  FORMAT(1X,'TOP RIGHT CORNER')
      GO TO 470
 470  CONTINUE
      GO TO (471,472,473,474,475,476,477,478,479,480,481,482,
     & 483),IFLAG(K)
 471  WRITE(6,492)
 492  FORMAT(1H+,'                              CONST FLUX ON BOTH SIDES')
      GO TO 410
 472  WRITE(6,493)
 493  FORMAT(1H+,'                           CONVECTIVE ON BOTH SIDES')
      GO TO 410
 473  WRITE(6,494)
 494  FORMAT(1H+,'                             SEMI-INF ON BOTH SIDES')
      GO TO 410
 474  WRITE(6,484)
 484  FORMAT(1H+,'                              VERT CONST FLUX--HORIZ CONVECT')
      GO TO 410
 475  WRITE(6,485)
 485  FORMAT(1H+,'                           VERT CONST FLUX--HORIZ SEMI-INF')
      GO TO 410
 476  WRITE(6,486)
 486  FORMAT(1H+,'                               VERT CONVECT--HORIZ SEMI-INF')
      GO TO 410
 477  WRITE(6,487)
 487  FORMAT(1H+,'                              VERT CONVECT--HORIZ CONST FLUX')
      GO TO 410
 478  WRITE(6,488)
 488  FORMAT(1H+,'                             VERT SEMI-INF--HORIZ CONVECT')
      GO TO 410
 479  WRITE(6,489)
 489  FORMAT(1H+,'                            VERT SEMI-INF--HORIZ CONST FLUX')
      GO TO 410
 480  WRITE(6,491)
 491  FORMAT(1H+,'                                CONSTANT TEMPERATURE')
      GO TO 410
 481  CONTINUE
      WRITE(6,416)
      GO TO 410
 482  CONTINUE
      WRITE(6,417)
      GO TO 410
 483  CONTINUE
      WRITE(6,418)
      GO TO 410
 416  FORMAT(1H+,'                        SEMI-INFINITE')
 417  FORMAT(1H+,'                        CONVECTIVE SURFACE')
 418  FORMAT(1H+,'                        CONSTANT HEAT FLUX')
 410  CONTINUE
      WRITE(1,183) Q
      WRITE(6,183) Q
 183  FORMAT(/,1X,'NUMBER OF TIME STEPS=',I5)
      DD=Q*DELT
      RRX=X/17
      JJX=RRX
      JJR=X-(17*JJX)
C
C     WRITE THE FINAL TEMPS & THERMAL VALUES
      DO 510 KK=1,5
      GO TO (511,512,513,514,515),KK
 511  WRITE(6,893) Q,DD
      GO TO 516
 512  WRITE(6,911)
      GO TO 516
 513  WRITE(6,912)
      GO TO 516
 514  WRITE(6,914)
      GO TO 516
 515  WRITE(6,913)
      GO TO 516
 516  DO 518 JJ=1,JJX
      IF(JJX.EQ.0) GO TO 518
      J2=17*JJ
      J1=J2-16
      GO TO (519,520,521,522,523),KK
 519  WRITE(6,500) ((RAY(I,J,1),J=J1,J2),I=1,Y)
      GO TO 517
 520  WRITE(6,500) ((TK(I,J),J=J1,J2),I=1,Y)
      GO TO 517
 521  WRITE(6,500) ((CP(I,J),J=J1,J2),I=1,Y)
      GO TO 517
```

```
  522 WRITE(6,500) ((RO(I,J),J=J1,J2),I=1,Y)
      GO TO 517
  523 WRITE(6,502) ((ISTAT(I,J),J=J1,J2),I=1,Y)
  517 IF(JJ.LT.JJX) WRITE(6,503)
  518 CONTINUE
      J2=X
      J1=X-JJR+1
      IF(JJR.EQ.0) GO TO 524
      IF(JJX.NE.0) WRITE(6,503)
      DO 524 I=1,Y
      GO TO (525,526,527,528,529),KK
  525 WRITE(6,500) (RAY(I,J,1),J=J1,J2)
      GO TO 524
  526 WRITE(6,500) (TK(I,J),J=J1,J2)
      GO TO 524
  527 WRITE(6,500) (CP(I,J),J=J1,J2)
      GO TO 524
  528 WRITE(6,500) (RO(I,J),J=J1,J2)
      GO TO 524
  529 WRITE(6,502) (ISTAT(I,J),J=J1,J2)
  524 CONTINUE
  510 CONTINUE
  500 FORMAT(1X,17F7.2)
  911 FORMAT(/,1X,'FINAL TK(I,J):')
  912 FORMAT(/,1X,'FINAL CP(I,J):')
  501 FORMAT(1X,17F7.2)
  914 FORMAT(/,1X,'FINAL RO(I,J):')
  913 FORMAT(/,1X,'FINAL ISTAT(I,J):')
  502 FORMAT(1X,17I3)
  503  FORMAT(/,3X,'AND THE NEXT SET OF COLUMNS:')
      WRITE(11,915) CPPC(1),ROPC(1)
      WRITE(11,916) TPC(1),TDEL
  915 FORMAT(/,1X,'CPPC(1)=',F6.2,'  ROPC(1)=',F6.2)
  916 FORMAT(1X,'TPC(1)=',F6.2,'   TDEL=',F6.2)
 9998  CONTINUE
  927  FORMAT(/,/)
C          CREATE A NEW DATA FILE WITH END RESULTS
      WRITE(9,10) DS,DELT,DI,TDEL
      WRITE(9,20) A,X,Y,NISO,ITRT,IMAX,ITPC
      WRITE(9,16) ((ROPC(K),CPPC(K),HL(K),TPC(K)),K=1,A)
      WRITE(9,15)(TISO(B),B=1,NISO)
      WRITE(9,30) (H(L),L=1,A)
      WRITE(9,35) ((RAY(I,J,3),J=1,X),I=1,Y)
      WRITE(9,35) ((RAY(I,J,2),J=1,X),I=1,Y)
      WRITE(9,35) ((RAY(I,J,1),J=1,X),I=1,Y)
      WRITE(9,34) (KRNR(J),J=1,4)
      WRITE(9,35) (FLXT(J),J=1,X)
      WRITE(9,35) (FLXB(J),J=1,X)
      WRITE(9,35) (FLXL(I),I=1,Y)
      WRITE(9,35) (FLXR(I),I=1,Y)
      WRITE(9,35) (TMPT(J),J=1,X)
      WRITE(9,35) (TMPB(J),J=1,X)
      WRITE(9,35) (TMPL(I),I=1,Y)
      WRITE(9,35) (TMPR(I),I=1,Y)
 9997 FORMAT(1X,'NO EQUATION FOR RAY(',I2,',',I2,').')
 9999  CONTINUE
C        CLOSE FILES
      CALL CONTRL(4,'ADPTMP',11)
      CALL CONTRL(4,'ADPDAT',5)
      CALL CONTRL(4,'ADPOUT',6)
      CALL CONTRL(4,'ADPNDT',9)
      CALL CONTRL(4,'POINT1',13)
      CALL EXIT
      END
C ------------------------------------------------------------------
C ******************************************************************
      SUBROUTINE TRIDIG(N,A,B,C,D)
      IMPLICIT INTEGER*2(I,N)
      IMPLICIT DOUBLE PRECISION(A,B,C,D)
      DIMENSION A(1),B(1),C(1),D(1)
C          ADJUST COEFFS FOR B&D FROM ELIMINATING A
      DO 10 I=2,N
      AB=A(I)/B(I-1)
      B(I)=B(I)-AB*C(I-1)
      D(I)=D(I)-AB*D(I-1)
   10 CONTINUE
C           BACK SUBSTITUTE
      N1=N-1
      D(N)=D(N)/B(N)
      DO 20 I=1,N1
      M=N-I
      D(M)=(D(M)-C(M)*D(M+1))/B(M)
   20 CONTINUE
      RETURN
      END
C ------------------------------------------------------------------
C ****************************************
      SUBROUTINE ISOTHM
C  THIS PROGRAM FINDS ISOTHERMS IN RAY(I,J,1).
      IMPLICIT INTEGER*2(A,B,I-L,N,Q,W,X,Y,Z)
      IMPLICIT DOUBLE PRECISION(C-H,M,O,P,R-V)
      DIMENSION EXRY(9,200), EYRY(9,200),COUNT(9)
      COMMON/M12I/ X,Y,NISO,Q
```

```
      COMMON/M12R/ RAY(80,80,3),TISO(9),DS,DELT
C    NISO=NUMBER OF ISOTHERMS
C    COUNT(B)=COUNTER FOR NO. ELTS IN EACH ISOTHERM
C    TISO(B)=TEMP OF EACH ISOTHERM
C            LET TISO(1) BE THE HOTTEST ISOTHERM
C    EXRY(B,K)= ARRAY FOR X-COORDINATES
C          B INDICATES WHICH ISOTHERM
C          K=COUNT(B) & INDICATES POSITION OF PT IN ISOTHERM LIST
C
C
C    CHANGE THE NEXT STATEMENT TO AGREE WITH DIMENSION
      DO 6 K=1,200
      DO 7 B=1,NISO
      EXRY(B,K)=0
      EYRY(B,K)=0
 7    CONTINUE
 6    CONTINUE
      DO 8 K=1,9
      COUNT(K)=0.
 8    CONTINUE
      CNTPT=0.
C    LOOP TO LOCATE ISOTHERMS
C           FOR LEFT SEMI-INNF,LET XL=3
C           FOR BOTTOM SEMI-INF,LET YB=Y-3
C           FOR RIGHT SEMI-INF,LET XR=X-3, AND COMMENT OUT LOOP 800
C           FOR TOP SEMI-INF,LET YT=3, AND COMMENT OUT LOOP 860
C            IF NOT USING SEMI-INF,XL=1,XR=X-1,YT=2,YB=Y
      XL=1
      XR=X-1
      YB=Y-3
      YT=2
      DO 100 I=YT,YB
      DO 200 J=XL,XR
C
C EXAMINE TEMPS HORIZONTALLY
      RJ=RAY(I,J,1)
      RJ1=RAY(I,(J+1),1)
      IF((RJ.GT.TISO(1)).AND.(RJ1.GT.TISO(1))) GO TO 500
      IF((RJ.LT.TISO(NISO)).AND.(RJ1.LT.TISO(NISO))) GO TO 500
      DO 500 B=1,NISO
      IF((TISO(B).GT.RJ).AND.(TISO(B).LT.RJ1)) GO TO 400
      IF((TISO(B).LT.RJ).AND.(TISO(B).GT.RJ1)) GO TO 400
      IF(TISO(B).EQ.RJ) GO TO 402
      GO TO 500
 400  CONTINUE
 401  FORMAT(1X,'RAY(',I3,I3,',1)=',F6.2)
      EXCD=(RAY(I,J,1)-TISO(B))/(RAY(I,J,1)-RAY(I,(J+1),1))+J
      EXCO=DS*(EXCD-1.)
      COUNT(B)=COUNT(B)+1
      K=COUNT(B)
      EXRY(B,K)=EXCO
      EYRY(B,K)=DS*(I-1.)
      CNTPT=CNTPT+1
      GO TO 500
 402  CONTINUE
      COUNT(B)=COUNT(B)+1
      K=COUNT(B)
      EXRY(B,K)=DS*(J-1.)
      EYRY(B,K)=DS*(I-1.)
      CNTPT=CNTPT+1
 500  CONTINUE
C EXAMINE TEMPS VERTICALLY
      RI=RAY(I,J,1)
      RI1=RAY((I-1),J,1)
      IF((RI.GT.TISO(1)).AND.(RI1.GT.TISO(1))) GO TO 525
      IF((RI.LT.TISO(NISO)).AND.(RI1.LT.TISO(NISO))) GO TO 525
      DO 525 B=1,NISO
      IF((TISO(B).GT.RI).AND.(TISO(B).LT.RI1)) GO TO 550
      IF((TISO(B).LT.RI).AND.(TISO(B).GT.RI1)) GO TO 550
      GO TO 525
 550  CONTINUE
 404  FORMAT(1X,'RJ=',F6.2,'      RJ1=',F6.2)
      EYCD=(RAY((I-1),J,1)-TISO(B))/(RAY((I-1),J,1)-
     & RAY(I,J,1))+(I-1.)
      EYCO=DS*(EYCD-1.)
      COUNT(B)=COUNT(B)+1
      K=COUNT(B)
      EYRY(B,K)=EYCO
      EXRY(B,K)=DS*(J-1.)
      CNTPT=CNTPT+1
 525  CONTINUE
 200  CONTINUE
 100  CONTINUE
C  800 LOOP IS FOR RIGHT HAND SIDE
      DO 800 I=YT,YB
      DO 851 B=1,NISO
      RI=RAY(I,X,1)
      RI1=RAY((I-1),X,1)
      IF((RI.GT.(TISO(B))).AND.(RI1.LT.(TISO(B)))) GO TO 852
      IF((RI.LT.(TISO(B))).AND.(RI1.GT.(TISO(B)))) GO TO 852
      IF(RI.EQ.TISO(B)) GO TO 853
```

```
      GO TO 851
852   CONTINUE
      EYCD=(I-1.)+(RI1-TISO(B))/(RI1-RAY(I,X,1))
      EYCO=DS*(EYCD-1.)
      COUNT(B)=COUNT(B)+1
      K=COUNT(B)
      EYRY(B,K)=EYCO
      EXRY(B,K)=DS*(X-1.)
      CNTPT=CNTPT+1
      GO TO 851
853   CONTINUE
      COUNT(B)=COUNT(B)+1
      K=COUNT(B)
      EYRY(B,K)=DS*(I-1.)
      EXRY(B,K)=DS*(X-1.)
      CNTPT=CNTPT+1
851   CONTINUE
800   CONTINUE
C            860 LOOP IS FOR TOP ROW
      XX=X-1
      I=1
      DO 860 J=XL,XR
      RJ=RAY(1,J,1)
      RJ1=RAY(1,(J+1),1)
      DO 861 B=1,NISO
      IF((TISO(B).GT.RJ).AND.(TISO(B).LT.RJ1)) GO TO 862
      IF((TISO(B).LT.RJ).AND.(TISO(B).GT.RJ1)) GO TO 862
      IF(RJ.EQ.TISO(B)) GO TO 863
      GO TO 861
862   CONTINUE
      EXCD=(RAY(I,J,1)-TISO(B))/(RAY(I,J,1)-RAY(I,(J+1),1))+J
      EXCO=DS*(EXCD-1.)
      COUNT(B)=COUNT(B)+1
      K=COUNT(B)
      EXRY(B,K)=EXCO
      EYRY(B,K)=0
      CNTPT=CNTPT+1
      GO TO 861
863   CONTINUE
      COUNT(B)=COUNT(B)+1
      K=COUNT(B)
      EXRY(B,K)=DS*(J-1.)
      EYRY(B,K)=0
      CNTPT=CNTPT+1
861   CONTINUE
860   CONTINUE
C
C
      TIM=Q*DELT
      WRITE(13,96) CNTPT,TIM
96    FORMAT(1X,'THE FOLLOWING',I5,' POINTS REPRESENT TIME=',
     & F8.4,' HOURS:')
94    FORMAT(1X,F7.2)
92    FORMAT(1X,I5)
      DO 666 B=1,NISO
      L=COUNT(B)
      IF(L.EQ.0) GO TO 99
      WRITE(13,91) L,B,TISO(B)
91    FORMAT(1X,2I6,F6.2)
99    CONTINUE
      DO 95 K=1,L
      IF(L.EQ.0.) GO TO 95
      WRITE(13,90) (EXRY(B,K),EYRY(B,K))
95    CONTINUE
90    FORMAT(1X,2F15.6)
666   CONTINUE
93    FORMAT(1X,'TOTAL NO. POINTS FOUND=',I5)
      RETURN
      END
C *****************************************************
C *********************************************************
C ADDATA, DATA-GATHERING SUBROUTINE
      SUBROUTINE ADDATA
      IMPLICIT INTEGER*2(A,B,I-L,N,Q,W,X,Y,Z)
      IMPLICIT DOUBLE PRECISION(C-H,M,O,P,R-V)
      COMMON/M12I/ X,Y,NISO,Q
      COMMON/M12R/ RAY(80,80,3),TISO(9),DS,DELT
      COMMON/M1I/ IMAX,A,ITRT,KRNR(4),ITPC
      COMMON/M1R/ DI,TDEL,H(2),
     & FLXT(80),FLXB(80),FLXL(80),FLXR(80),TMPT(80),
     & TMPB(80),TMPL(80),TMPR(80),ROPC(2),CPPC(2),HL(2),TPC(2)
C     ADPDAT IS THE DATA FILE FOR ADIPC
C            ALL UNITS ARE IN METERS,HOURS,CELSIUS
C     DELT=TIME INCREMENT (HRS)
C     IMAX=MAX NO. OF TIME STEPS REQUESTED
C     DS=DISTANCE BETWEEN NODES (METERS)
C     DI=THE MULTIPLE OF DS WHICH IS HALF THE DISTANCE TO INFINIY.
C        (DS*DI=HALF THE DISTANCE TO INFINITY)  DI>1
C     TSRF=SURFACE TEMP (OUTSIDE GRID) (INFINITE DIST AWAY)
C     TBTM=BOTTOM TEMP (UNDER GRID) (INFINITE DIST AWAY)
C     TRIT=TEMP TO RIGHT OF GRID (INFINITE DIST)
C     TLFT=TEMP TO LEFT OF GRID  (INFINITE DIST AWAY)
C     X= NO. OF GRID NODES HORIZONTALLY (X>=2)
C     Y= NO. OF GRID NODES VERTICALLLY (Y>=2)
C     A= NO. OF DIFFERENT MATERIALS IN THE GRID
```

```
C     CP(I,J)=SPECIFIC HEAT    (W*HR/KG*K)
C     TK(I,J)=THERMAL CONDUCTIVITY    (W/M*K)
C     RO(I,J)=DENSITY    (KG/M3)
C     H(L)=CONVECTION COEFFICIENT
C      CPO=SPECIFIC HEAT OF PREVIOUS TIME STEP (FIGURED IN MAIN PROGRAM)
C      ROO=DENSITY OF PREVIOUS TIME STEP (FIGURED IN MAIN PROGRAM)
C     RAY(I,J,1)=PRESENT NODAL TEMP
C     RAY(I,J,2)=LOCATION TYPE:
C               THIS PROGRAM HAS SEMI-INFINITE OPTION ON LEFT,
C               BOTTOM, AND RIGHT SIDES ONLY, AT PRESENT.
C               ASTERISKS DENOTE SITUATIONS NOT YET PROGRAMMED.
C              =1 TOP LEFT CORNER
C              =2 BOTTOM LEFT CORNER
C              =3 BOTTOM RIGHT CORNER
C              =4 TOP RIGHT CORNER
C              =5 INTERIOR NODE, FLUCTUATING TEMP
C              =6 INTERIOR NODE,CONSTANT TEMP
C              =7 NODE ON CONST TEMP BOUNDARY
C              =8  CONST FLUX BOUNDARY: TOP
C              =27 CONST FLUX BOUNDARY: LEFT
C              =28 CONST FLUX BOUNDARY: BOTTOM
C              =29 CONST FLUX BOUNDARY: RIGHT
C              =9  CONVECTIVE BOUNDARY: TOP
C              =30 CONVECTIVE BOUNDARY: LEFT
C              =31 CONVECTIVE BOUNDARY: BOTTOM
C              =32 CONVECTIVE BOUNDARY: RIGHT
C              =10 NODE ON SEMI-INFINITE BOUNDARY
C              =11     NODE ADJACENT TO LEFT SEMI-INF BNDRY
C              =12     NODE ADJACENT TO BOTTOM SEMI-INF BNDARY
C              =13     NODE ADJACENT TO RIGHT SEMI-INF BNDRY
C              =14     NODE ADJACENT TO TOP SEMI-INF BNDRY *
C              THE NEXT FIVE CONDITIONS NEEDED WITH SEMI-INF CORNER
C              =15      SQUARE NODE ADJACENT TO TWO SEMI-INF SIDES
C               =16    SEMI-INF NODE ABOVE SEMI-INF CORNER NODE
C                        (FOR BOTTOM LEFT & RIGHT ONLY, AT PRESENT)
C              =17     SEMI-INF NODE TO LEFT OF BOTTOM RIGHT SEMI-INF CORNER NODE
C              =18     SEMI-INF NODE BELOW SEMI-INF CORNER NODE *
C              =19     SEMI-INF NODE TO RIGHT OF BOTTOM LEFT SEMI-INF CORNER NODE
C               THE NEXT FOUR CONDTNS NEEDED WITH CORNER W ONE SIDE SEMI-INF,
                  ONE SIDE CONST FLUX
C              =20     TOP CONST FLUX NODE ADJACENT TO RIGHT SEMI-INF
C              =21     RIGHT SIDE CONST FLUX NODE ADJACENT TO BOTTOM SEMI-INF
C              =22     TOP CONST FLUX NODE ADJACENT TO LEFT SEMI-INF
C              =23     LEFT SIDE CONST FLUX NODE ADJACENT TO BOTTOM SEMI-INF
C              THE NEXT THREE CONDTNS NEEDED WITH CORNER W ONE SIDE SEMI-INF,
                 ONE SIDE CONVECTIVE
C              =24    RIGHT SIDE CONVECT NODE ADJ TO BOTTOM SEMI-INF*
C              =25     TOP CONVECT NODE ADJ TO LEFT SEMI-INF*
C              =26     LEFT SIDE CONST FLUX NODE ADJACENT TO BOTTOM SEMI-INF*
C     RAY(I,J,3)= INDEX OF MATERIAL (RANGES FROM 1 TO A)
C     KRNR(1)=TOP LEFT CORNER
C     KRNR(2)=BOTTOM LEFT CORNER
C     KRNR(3)=BOTTOM RIGHT CORNER
C     KRNR(4)=TOP RIGHT CORNER
C     KRNR(L)=1 CONSTANT TEMP CORNER NODE
C            =2 CONSTANT FLUX (ON BOTH SIDES) CORNER NODE
C            =3 CONVECTIVE (ON BOTH SIDES) CORNER NODE
C            =4 SEMI-INFINITE (ON BOTH SIDES) CORNER NODE
C            =5 VERTICAL SIDE=CONVECT, HORIZ=CONST FLUX
C            =6 VERTICAL SIDE=CONST FLUX, HORIZ =CONVECT
C            =7 VERTICAL SIDE =CONST FLUX, HORIZ =SEMI-INF
C            =8 VERTICAL SIDE=SEMI-INF, HORIZ=CONST FLUX
C                (TOP LEFT & TOP RIGHT ONLY AT PRESENT)
C            =9 VERTICAL SIDE =CONVECT, HORIZ=SEMI-INF *
C            =10 VERTICAL SIDE=SEMI-INF, HORIZ=CONVECT*
C     TEMP(I,J)= TEMP AT TIME (T+DELT)
C     ISTAT(I,J)=INDEX OF STATE FOR TEMP(I,J)
C              =1 FROZEN
C              =2 UNFROZEN
C              =3 UNDERGOING PHASE CHANGE
C     FLXT(J)=FLUX FROM TOP OF GRID
C     FLXB(J)=FLUX FROM BOTTOM OF GRID
C     FLXL(I)=FLUX FROM LEFT SIDE OF GRID
C     FLXR(I)=FLUX FROM RIGHT SIDE OF GRID
C     TMPT(J)=TEMP OUTSIDE GRID FOR TOP
C     TMPB(J)=TEMP OUTSIDE GRID FOR BOTTOM
C     TMPL(I)=TEMP OUTSIDE GRID FOR LEFT
C     TMPR(I)=TEMP OUTSIDE GRID FOR RIGHT
C     NISO= NO. OF ISOTHERMS TO BE PLOTTED
C     TMAX= TEMP OF HOTTEST ISOTHERM
C     TMIN= TEMP OF COLDEST ISOTHERM
C     TISO(B)= ARRAY CONTAINING ISOTHERM TEMPS
C            TISO(1) HAS THE HOTTEST ISOTHERM
C     COUNT(B)=COUNTER FOR NO.ELTS IN EACH ISOTHERM
C     ITRT=NO. TIME STEPS BEFORE TEMPS PRINTED
C     ITPC=NO. TIME STEPS BEFORE ISOTHM IS CALLED
C          FOR THE NEXT 4 STMTS,K=RAY(I,J,3)
C     ROPC(K)=DENSITY AT PHASE CHANGE FRONT(KG/M3)
C     HL(K)=LATENT HEAT OF FUSION (W*HR/KG*K)
C     TPC(K)=TEMPERATURE OF PHASE CHANGE (*C)
C     CPPC(K)=SPECIFIC HEAT DURING PHASE CHANGE
C     TDEL=TOTAL TEMP RANGE FOR TPC
C          (EX: IF TPC=0*,TDEL=2,PHSCH OCCURS FROM -1* TO 1*)
```

```
C INITIALIZE VARIABLES AND ARRAYS
      TSRF=-4.670D0
      TBTM=4.67D0
      TLFT=4.670D0
      TRIT=4.670D0
      DS=.0050D0
      DELT=.00250D0
      DI=50.0D0
      A=1
      IMAX=1200
      X=3
      Y=25
      TDEL=1.0D0
      NISO=1
      ITRT=40
      ITPC=10
C
C DO NOT CHANGE THE FOLLOWING 6 LINES.
      ETIME=ITRT*DELT
      MTIME=IMAX*DELT
      WRITE(1,3) ITRT,ETIME,IMAX,MTIME
  3   FORMAT(1X,'TEMPERATURES WILL BE PRINTED EVERY',I3,' ITNS',/,
     C 1X,'  (EVERY',F12.4,' HRS.)',/,1X,
     C 'MAX NO. ITNS=',I4,'    (=',F12.4,' HRS)')
C
C SET UP MATERIAL PROPERTIES FOR EACH MATERIAL.
C   (YOU SET THE VALUES OF K & CP & RO IN MAIN PROGRAM.)
      H(A)=0.D0
C     PROPERTIES FOR PHASE CHANGE:
      ROPC(1)=917.0D0
      HL(1)=93.0D0
      TPC(1)=0.0D0
C     THE FOLLOWING SPECIFIC HEAT SHOULD INCLUDE LATENT HEAT.
C          YOU CALCULATE IT BY: CP=(CP(FROZ)+CP(UNFROZ))/2 + HL(K)/TDEL
      DO 2 K=1,A
      CPPC(K)=093.870D0
  2   CONTINUE
C
C SPECIFY THE ISOTHERMS TO BE LOCATED IN SUBROUTINE ISOTHM
C   IN ORDER, WITH TISO(1) THE HOTTEST.
      TISO(1)=0.0D0
C
C INDICATE THE FLUXES FROM THE SIDES OF THE GRID.
C       THIS IS USED ONLY FOR THE CONSTANT FLUX BOUND.CONDITION.
      DO 11 J=1,X
      FLXT(J)=0.D0
      FLXB(J)=0.D0
   11   CONTINUE
      DO 12 I=1,Y
      FLXL(I)=0.D0
      FLXR(I)=0.D0
  12  CONTINUE
C
C INDICATE THE TEMPERATURES OUTSIDE THE GRID.
C        USED FOR SEMI-INF & CONVECT BOUNDARYS
      DO 13 J=1,X
      TMPT(J)=TSRF
      TMPB(J)=TBTM
  13  CONTINUE
      DO 14 I=1,Y
      TMPL(I)=TLFT
      TMPR(I)=TRIT
  14  CONTINUE
C
C SET UP RAY(I,J,3) INDEX OF MATERIALS
C    THERE ARE A MATERIALS. LET THE SURROUNDING
C    MATERIAL BE THE 'ATH' MATERIAL.  START
C    WITH '1'.
C THIS LOOP ASSIGNS THE 'ATH' MATERIAL TO
C    THE REMAINING NODES.
      DO 10 J=1,X
      DO 20 I=1,Y
      RAY(I,J,3)=A
  20  CONTINUE
  10  CONTINUE
C
C SET UP RAY(I,J,2) NODAL LOCATION TYPE
C    TOP BOUNDARY
      DO 40 J=1,X
      RAY(1,J,2)=7
  40  CONTINUE
C    BOTTOM BOUNDARY
      DO 50 J=1,X
      RAY(Y,J,2)=10
  50  CONTINUE
C    LEFT BOUNDARY
      DO 60 I=1,Y
      RAY(I,1,2)=27
  60  CONTINUE
C    RIGHT BOUNDARY
      DO 70 I=1,Y
      RAY(I,X,2)=29
  70  CONTINUE
C SET UP CORNER CONDITIONS
```

```
C   --DON'T CHANGE THE "RAY(I,J,2)" STATEMENTS.
      RAY(1,1,2)=1
      RAY(Y,1,2)=2
      RAY(Y,X,2)=3
      RAY(1,X,2)=4
C           TOP LEFT CORNER
      KRNR(1)=1
      RAY(1,1,1)=TSRF
C            BOTTOM LEFT CORNER
      KRNR(2)=7
      RAY(Y,1,1)=TBTM
C            BOTTOM RIGHT CORNER
      KRNR(3)=7
      RAY(Y,X,1)=TBTM
C          TOP RIGHT CORNER
      KRNR(4)=1
      RAY(1,X,1)=TSRF
C
C INTERIOR NODES GIVEN FLUCTUATING TEMP
C  DO NOT CHANGE THE 80,90 LOOP.
      YY=Y-1
      XX=X-1
      DO 80 J=2,XX
      DO 90 I=2,YY
      RAY(I,J,2)=5
 90   CONTINUE
 80   CONTINUE
C
C CONSTANT INTERIOR NODES.
C  INDICATE RAY(I,J,2)=6 IF RELEVANT
C
C IF RELEVENT, SPECIFY SPECIAL RAY(I,J,2) CONDITIONS
C       TO BE USED WITH SEMI-INF BOUNDARY.
      XX=X-1
      YY=Y-1
      DO 210 J=2,XX
      RAY(YY,J,2)=12
 210  CONTINUE
C     RAY(YY,2,2)=15
      RAY(YY,1,2)=23
C     RAY(Y,2,2)=19
C     RAY(YY,XX,2)=15
      RAY(YY,X,2)=21
C
C SET UP RAY(I,J,1) PRESENT NODAL TEMP
C       DO NOT ALTER THIS
      XX=X-1
      YY=Y-1
      DO 110 J=2,XX
      RAY(1,J,1)=TSRF
      RAY(Y,J,1)=TBTM
 110  CONTINUE
      DO 120 I=2,YY
      RAY(I,1,1)=TLFT
      RAY(I,X,1)=TRIT
 120  CONTINUE
      XX=X-1
      YY=Y-1
      DO 130 J=2,XX
      DO 140 I=2,YY
C   COMMENT OUT THE INITIAL TEMP DIST THAT YOU DON'T WANT:
C   NODES ASSIGNED LINEAR TEMP DIST VERTICALLY
C     RAY(I,J,1)=TSRF-((TSRF-TBTM)/(Y-1))*(I-1)
C   NODES ASSIGNED UNIFORM TEMP (INDICATE THE TEMP:)
      RAY(I,J,1)=4.670D0
 140  CONTINUE
 130  CONTINUE
C
C INTERIOR NODES
C    INDICATE THE TEMPERATURES FOR RAY(I,J,1)
C        IF THEY ARE DIFFERENT THAN ASSIGNED IN THE 130,140 LOOP.
C
C
C   DO NOT CHANGE THE FOLLOWING 17 STATEMENTS.
      WRITE(5,131) DS,DELT,DI,TDEL
 131  FORMAT(1X,4F10.5)
      WRITE(5,132) A,X,Y,NISO,ITRT,IMAX,ITPC
 132  FORMAT(1X,7I5)
      WRITE(5,136) ((ROPC(K),CPPC(K),HL(K),TPC(K)),K=1,A)
 136  FORMAT(1X,4F10.5)
      WRITE(5,134)(TISO(B),B=1,NISO)
 134  FORMAT(1X,F7.2)
      WRITE(5,133) (H(L),L=1,A)
 133  FORMAT(1X,F10.5)
      WRITE(5,151) ((RAY(I,J,3),J=1,X),I=1,Y)
      WRITE(5,151) ((RAY(I,J,2),J=1,X),I=1,Y)
      WRITE(5,151) ((RAY(I,J,1),J=1,X),I=1,Y)
      WRITE(5,152) (KRNR(J),J=1,4)
 152  FORMAT(1X,4I2)
 151  FORMAT(1X,17F7.2)
      WRITE(5,151) (FLXT(J),J=1,X)
      WRITE(5,151) (FLXB(J),J=1,X)
      WRITE(5,151) (FLXL(I),I=1,Y)
      WRITE(5,151) (FLXR(I),I=1,Y)
```

```
      WRITE(5,151) (TMPT(J),J=1,X)
      WRITE(5,151) (TMPB(J),J=1,X)
      WRITE(5,151) (TMPL(I),I=1,Y)
      WRITE(5,151) (TMPR(I),I=1,Y)
      RETURN
      END
C ******************************************************
```

Data file ADPDAT printed by subroutine ADDATA for the Neumann problem

```
   0.00500     0.00250   50.00000     1.00000
   1      3    25      1    40 1200   10
 917.00000   93.87000   93.00000     0.00000
   0.00
   0.00000
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   1.00   1.00   1.00   1.00   1.00   1.00
   1.00   7.00   4.00  27.00   5.00  29.00  27.00   5.00  29.00  27.00   5.00  29.00  27.00   5.00  29.00  27.00   5.00
  29.00  27.00   5.00  29.00  27.00   5.00  29.00  27.00   5.00  29.00  27.00   5.00  29.00  27.00   5.00  29.00  27.00
   5.00  29.00  27.00   5.00  29.00  27.00   5.00  29.00  27.00   5.00  29.00  27.00   5.00  29.00  27.00   5.00  29.00
  27.00   5.00  29.00  27.00   5.00  29.00  27.00   5.00  29.00  27.00   5.00  29.00  27.00   5.00  29.00  27.00   5.00
  29.00  23.00  12.00  21.00   2.00  10.00   3.00
  -4.67  -4.67  -4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67
   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67
   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67
   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67
   4.67   4.67   4.67   4.67   4.67   4.67   4.67
 1 7 7 1
   0.00   0.00   0.00
   0.00   0.00   0.00
   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
  -4.67  -4.67  -4.67
   4.67   4.67   4.67
   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67
   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67
   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67
   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67
```

File ADPOUT

(Lists the initial data in readable form and the final values from the run [printed by ADIPC].)

```
DATA FOR THIS RUN OF ADIPC:

     DS          DELT          DI          TDEL
   0.00500     0.00250     50.00000     1.00000

     A      X      Y  NISO  ITRT  IMAX  ITPC
     1      3     25     1     40  1200    10

   ROPC(K)      CPPC(K)      HL(K)       TPC(K)
  917.00000    93.87000    93.00000     0.00000

TISO(B),B=1,NISO:
   0.00

   H(K)
   0.00000

KRNR(1)= 1  KRNR(2)= 7   KRNR(3)= 7   KRNR(4)= 1

   FLXT(J)      FLXB(J)      TMPT(J)      TMPB(J)
      0.00         0.00        -4.67         4.67
      0.00         0.00        -4.67         4.67
      0.00         0.00        -4.67         4.67

   FLXL(I)      FLXR(I)      TMPL(I)      TMPR(I)
      0.00         0.00         4.67         4.67
      0.00         0.00         4.67         4.67
      0.00         0.00         4.67         4.67
      0.00         0.00         4.67         4.67
      0.00         0.00         4.67         4.67
      0.00         0.00         4.67         4.67
      0.00         0.00         4.67         4.67
      0.00         0.00         4.67         4.67
      0.00         0.00         4.67         4.67
      0.00         0.00         4.67         4.67
      0.00         0.00         4.67         4.67
      0.00         0.00         4.67         4.67
      0.00         0.00         4.67         4.67
      0.00         0.00         4.67         4.67
      0.00         0.00         4.67         4.67
      0.00         0.00         4.67         4.67
      0.00         0.00         4.67         4.67
      0.00         0.00         4.67         4.67
      0.00         0.00         4.67         4.67
      0.00         0.00         4.67         4.67
      0.00         0.00         4.67         4.67
      0.00         0.00         4.67         4.67
      0.00         0.00         4.67         4.67
      0.00         0.00         4.67         4.67
      0.00         0.00         4.67         4.67

RAY(I,J,1)    TEMPERATURES AT THE START:
   -4.67    -4.67    -4.67
    4.67     4.67     4.67
    4.67     4.67     4.67
    4.67     4.67     4.67
    4.67     4.67     4.67
    4.67     4.67     4.67
    4.67     4.67     4.67
    4.67     4.67     4.67
    4.67     4.67     4.67
    4.67     4.67     4.67
    4.67     4.67     4.67
    4.67     4.67     4.67
    4.67     4.67     4.67
    4.67     4.67     4.67
    4.67     4.67     4.67
    4.67     4.67     4.67
    4.67     4.67     4.67
    4.67     4.67     4.67
    4.67     4.67     4.67
    4.67     4.67     4.67
    4.67     4.67     4.67
    4.67     4.67     4.67
    4.67     4.67     4.67
    4.67     4.67     4.67
    4.67     4.67     4.67

RAY(I,J,2)    NODAL LOCATION TYPE
    1.00     7.00     4.00
   27.00     5.00    29.00
   27.00     5.00    29.00
   27.00     5.00    29.00
   27.00     5.00    29.00
```

```
  27.00    5.00   29.00
  27.00    5.00   29.00
  27.00    5.00   29.00
  27.00    5.00   29.00
  27.00    5.00   29.00
  27.00    5.00   29.00
  27.00    5.00   29.00
  27.00    5.00   29.00
  27.00    5.00   29.00
  27.00    5.00   29.00
  27.00    5.00   29.00
  27.00    5.00   29.00
  27.00    5.00   29.00
  27.00    5.00   29.00
  27.00    5.00   29.00
  27.00    5.00   29.00
  27.00    5.00   29.00
  27.00    5.00   29.00
  23.00   12.00   21.00
   2.00   10.00    3.00
RAY(I,J,3)    NODAL MATERIAL TYPE
   1.00    1.00    1.00
   1.00    1.00    1.00
   1.00    1.00    1.00
   1.00    1.00    1.00
   1.00    1.00    1.00
   1.00    1.00    1.00
   1.00    1.00    1.00
   1.00    1.00    1.00
   1.00    1.00    1.00
   1.00    1.00    1.00
   1.00    1.00    1.00
   1.00    1.00    1.00
   1.00    1.00    1.00
   1.00    1.00    1.00
   1.00    1.00    1.00
   1.00    1.00    1.00
   1.00    1.00    1.00
   1.00    1.00    1.00
   1.00    1.00    1.00
   1.00    1.00    1.00
   1.00    1.00    1.00
   1.00    1.00    1.00
   1.00    1.00    1.00
   1.00    1.00    1.00
   1.00    1.00    1.00

TEMPERATURES WILL BE PRINTED EVERY  40 TIME STEPS.
ISOTHERMS WILL BE LOCATED EVERY  10 TIME STEPS.

 BOUNDARY CONDITIONS:

TOP BOUNDARY                  CONSTANT TEMPERATURE
TOP LEFT CORNER               CONSTANT TEMPERATURE
LEFT BOUNDARY    CONSTANT HEAT FLUX
BOTTOM LEFT CORNER VERT CONST FLUX--HORIZ SEMI-INF
BOTTOM BOUNDARY SEMI-INFINITE
BOTTOM RIGHT CORNERVERT CONST FLUX--HORIZ SEMI-INF
RIGHT BOUNDARY   CONSTANT HEAT FLUX
TOP RIGHT CORNER              CONSTANT TEMPERATURE

NUMBER OF TIME STEPS= 1200


TEMPERATURES AFTER1200 TIME STEPS (        3.000 HRS):
  -4.67   -4.67   -4.67
  -3.74   -3.74   -3.74
  -2.82   -2.82   -2.82
  -1.89   -1.89   -1.89
  -0.97   -0.97   -0.97
  -0.04   -0.04   -0.04
   0.38    0.38    0.38
   0.65    0.65    0.65
   1.07    1.07    1.07
   1.46    1.46    1.46
   1.84    1.84    1.84
   2.18    2.18    2.18
   2.50    2.50    2.50
   2.79    2.79    2.79
   3.04    3.04    3.04
   3.27    3.27    3.27
   3.47    3.47    3.47
   3.64    3.64    3.64
   3.79    3.79    3.79
   3.91    3.91    3.91
   4.00    4.00    4.00
   4.07    4.07    4.07
   4.13    4.13    4.13
   4.16    4.16    4.16
   4.64    4.64    4.64
```

```
FINAL TK(I,J):
  2.21     2.21     2.21
  2.21     2.21     2.21
  2.21     2.21     2.21
  2.21     2.21     2.21
  2.21     2.21     2.21
  2.21     2.21     2.21
  2.21     2.21     2.21
  0.58     0.58     0.58
  0.58     0.58     0.58
  0.58     0.58     0.58
  0.58     0.58     0.58
  0.58     0.58     0.58
  0.58     0.58     0.58
  0.58     0.58     0.58
  0.58     0.58     0.58
  0.58     0.58     0.58
  0.58     0.58     0.58
  0.58     0.58     0.58
  0.58     0.58     0.58
  0.58     0.58     0.58
  0.58     0.58     0.58
  0.58     0.58     0.58
  0.58     0.58     0.58
  0.58     0.58     0.58
  0.58     0.58     0.58
```

```
FINAL CP(I,J):
  0.58     0.58     0.58
  0.58     0.58     0.58
  0.58     0.58     0.58
  0.58     0.58     0.58
  0.58     0.58     0.58
  1.16     1.16     1.16
  1.16     1.16     1.16
  1.16     1.16     1.16
  1.16     1.16     1.16
  1.16     1.16     1.16
  1.16     1.16     1.16
  1.16     1.16     1.16
  1.16     1.16     1.16
  1.16     1.16     1.16
  1.16     1.16     1.16
  1.16     1.16     1.16
  1.16     1.16     1.16
  1.16     1.16     1.16
  1.16     1.16     1.16
  1.16     1.16     1.16
  1.16     1.16     1.16
  1.16     1.16     1.16
  1.16     1.16     1.16
  1.16     1.16     1.16
  1.16     1.16     1.16
```

```
FINAL RO(I,J):
 917.00 917.00 917.00
 917.00 917.00 917.00
 917.00 917.00 917.00
 917.00 917.00 917.00
 917.00 917.00 917.00
 998.20 998.20 998.20
 998.20 998.20 998.20
 998.20 998.20 998.20
 998.20 998.20 998.20
 998.20 998.20 998.20
 998.20 998.20 998.20
 998.20 998.20 998.20
 998.20 998.20 998.20
 998.20 998.20 998.20
 998.20 998.20 998.20
 998.20 998.20 998.20
 998.20 998.20 998.20
 998.20 998.20 998.20
 998.20 998.20 998.20
 998.20 998.20 998.20
 998.20 998.20 998.20
 998.20 998.20 998.20
 998.20 998.20 998.20
 998.20 998.20 998.20
 998.20 998.20 998.20
```

```
FINAL ISTAT(I,J):
  1   1   1
  1   1   1
  1   1   1
  1   1   1
  1   1   1
  3   3   3
  3   3   3
  2   2   2
  2   2   2
  2   2   2
  2   2   2
  2   2   2
  2   2   2
  2   2   2
  2   2   2
  2   2   2
  2   2   2
  2   2   2
  2   2   2
  2   2   2
  2   2   2
  2   2   2
  2   2   2
  2   2   2
  2   2   2
```

Data file ADPNDT

(Contains the values from the end of the run [calculated by ADIPC] and could be used to start the program again to run for more time.)

```
  0.00500    0.00250  50.00000    1.00000
  1     3    25     1   40 1200   10
917.00000  93.87000  93.00000    0.00000
  0.00
  0.00000
  1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
  1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
  1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
  1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00   1.00
  1.00   1.00   1.00   1.00   1.00   1.00   1.00
  1.00   7.00   4.00  27.00   5.00  29.00  27.00   5.00  29.00  27.00   5.00  29.00  27.00   5.00  29.00  27.00   5.00
 29.00  27.00   5.00  29.00  27.00   5.00  29.00  27.00   5.00  29.00  27.00   5.00  29.00  27.00   5.00  29.00  27.00
  5.00  29.00  27.00   5.00  29.00  27.00   5.00  29.00  27.00   5.00  29.00  27.00   5.00  29.00  27.00   5.00  29.00
 27.00   5.00  29.00  27.00   5.00  29.00  27.00   5.00  29.00  27.00   5.00  29.00  27.00   5.00  29.00  27.00   5.00
 29.00  23.00  12.00  21.00   2.00  10.00   3.00
 -4.67  -4.67  -4.67  -3.74  -3.74  -3.74  -2.82  -2.82  -2.82  -1.89  -1.89  -1.89  -0.97  -0.97  -0.97  -0.04  -0.04
 -0.04   0.38   0.38   0.38   0.65   0.65   0.65   1.07   1.07   1.07   1.46   1.46   1.46   1.84   1.84   1.84   2.18
  2.18   2.18   2.50   2.50   2.50   2.79   2.79   2.79   3.04   3.04   3.04   3.27   3.27   3.27   3.47   3.47   3.47
  3.64   3.64   3.64   3.79   3.79   3.79   3.91   3.91   3.91   4.00   4.00   4.00   4.07   4.07   4.07   4.13   4.13
  4.13   4.16   4.16   4.16   4.64   4.64   4.64
1 7 7 1
  0.00   0.00   0.00
  0.00   0.00   0.00
  0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
  0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
  0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
  0.00   0.00   0.00   0.00   0.00   0.00   0.00   0.00
 -4.67  -4.67  -4.67
  4.67   4.67   4.67
  4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67
  4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67
  4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67
  4.67   4.67   4.67   4.67   4.67   4.67   4.67   4.67
```

Sample of isotherm locations output from ADIPC for the Neuman problem (file POINT1)

```
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.0250 HOURS:
     3       1   0.00
        0.000000           0.004632
        0.005000           0.004632
        0.010000           0.004632
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.0500 HOURS:
     3       1   0.00
        0.000000           0.004727
        0.005000           0.004727
        0.010000           0.004727
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.0750 HOURS:
     3       1   0.00
        0.000000           0.004830
        0.005000           0.004830
        0.010000           0.004830
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.1000 HOURS:
     3       1   0.00
        0.000000           0.004940
        0.005000           0.004940
        0.010000           0.004940
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.1250 HOURS:
     3       1   0.00
        0.000000           0.005197
        0.005000           0.005197
        0.010000           0.005197
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.1500 HOURS:
     3       1   0.00
        0.000000           0.005661
        0.005000           0.005661
        0.010000           0.005661
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.1750 HOURS:
     3       1   0.00
        0.000000           0.006161
        0.005000           0.006161
        0.010000           0.006161
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.2000 HOURS:
     3       1   0.00
        0.000000           0.006685
        0.005000           0.006685
        0.010000           0.006685
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.2250 HOURS:
     3       1   0.00
        0.000000           0.007223
        0.005000           0.007223
        0.010000           0.007223
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.2500 HOURS:
     3       1   0.00
        0.000000           0.009329
        0.005000           0.009329
        0.010000           0.009329
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.2750 HOURS:
     3       1   0.00
        0.000000           0.009413
        0.005000           0.009413
        0.010000           0.009413
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.3000 HOURS:
     3       1   0.00
        0.000000           0.009514
        0.005000           0.009514
        0.010000           0.009514
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.3250 HOURS:
     3       1   0.00
        0.000000           0.009618
        0.005000           0.009618
        0.010000           0.009618
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.3500 HOURS:
     3       1   0.00
        0.000000           0.009724
        0.005000           0.009724
        0.010000           0.009724
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.3750 HOURS:
     3       1   0.00
        0.000000           0.009831
        0.005000           0.009831
        0.010000           0.009831
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.4000 HOURS:
     3       1   0.00
        0.000000           0.009941
        0.005000           0.009941
        0.010000           0.009941
THE FOLLOWING      3 POINTS REPRESENT TIME=  0.4250 HOURS:
     3       1   0.00
        0.000000           0.010158
        0.005000           0.010158
        0.010000           0.010158
```

A facsimile catalog card in Library of Congress MARC format is reproduced below.

Albert, Mary Remley
Computer models for two-dimensional transient heat conduction / by Mary Remley Albert. Hanover, N.H.: Cold Regions Research and Engineering Laboratory; Springfield, Va.: available from National Technical Information Service, 1983.
iv, 74 p., illus.; 28 cm. (CRREL Report 83-12.)
Prepared for Office of the Chief of Engineers by Corps of Engineers, U.S. Army Cold Regions Research and Engineering Laboratory under DA Project 4A762730 AT42.
Bibliography: p. 30.
1. Centralized heating systems. 2. Computerized simulation. 3. Computer models. 4. Heat pipes. 5. Heat transfer. I. United States. Army. Corps of Engineers. II. Cold Regions Research and Engineering Laboratory, Hanover, N.H. III. Series: CRREL Report 83-12.

END

FILMED

12-83

DTIC